# The Development of a United States Medical Care Price Comparison Web Portal

**Team 4: John Parsons, Sam Glendenning, Youssef Aloulou, Victor Yu, Saif Fardan, Adam Munro**
**AC41004 Industrial Team Project**
**B.Sc. Hons Applied Computing / Computing Science**
**University of Dundee, 2019**
**Supervisor: Dr. C. Ramsay**

CompareTheHealthcare.com *is a medical healthcare price comparison website designed for citizens of the United States of America. It was designed and implemented by a team of six students at the University of Dundee for an external client experienced in the US healthcare market. Supported by a database of hundreds of medical procedures offered at thousands of hospitals in the US,* CompareTheHealthcare.com *offers information on the annual average prices for the Medicare-insured and uninsured citizen to compare between hundreds of cities across every state, providing the ability to find the most suitable and affordable healthcare for patients regardless of their location. During its three-week development lifecycle,* CompareTheHealthcare.com *underwent user testing and evaluations which concluded that the website offered a quick and efficient procedural search function on a clean and simplistic design suitable for end users of all web experience. It also concluded that there was work to be done in the data offered in the results, as well as clarification on the descriptions of each procedure and the risks associated with them.*

## 1 Introduction

“Health is a crown that the healthy wear, but only the sick can see” - as attributed to the Muslim scholar ash-Shafi’. Healthcare is fundamental: we are all in dire need to be healthy, to feel good, to know that treatment can be accessible to us at any time. Universal healthcare, that is, healthcare that is affordable by the majority of the country’s population, is a huge driving force in that notion, often a staple of a developed country that is taken for granted and which is seen by those with access to it as a boon.

With this in mind, it is surprising that such a Western power as the United States of America follows a different path: one in which citizens must pay for their own medical treatment. This is no small task. Modern medicine and healthcare has many large, unseen costs racking up to thousands of dollars for even one night in hospital. Because of this, the US is rich in supply of health insurers looking to make a buck or two out of something that the vast majority of American citizens need if they ever find themselves in the back of an ambulance.

Medical costs, no matter where you are in the world, are steep but highly variable on many factors and nowhere is this more plain than the United States. American citizens up and down the country are unaware of the savings they can make both when they have health insurance and when they shop around. The two main factors in Americans not having access to the highest quality and most affordable care are location and not searching for a good deal on your medical treatment. The same medical procedure can cost wildly different amounts depending on if it’s offered in Alaska or Arkansas. What’s more is that the same procedure can cost wildly different amounts even between neighbouring cities in the same state! And it’s true: you can potentially be paying tens of thousands of dollars more for your hospital treatment depending on whether your ambulance takes you to Florida Hospital in Orlando or Tampa General Hospital in Tampa.

For the average American citizen, this money matters and we believe Americans should stop paying more for treatment in one place when they have the potential to be going elsewhere for a lower price. Thus, this project’s objective is to help patients around the US to compare prices of procedures across different hospitals, whether insured or uninsured, allowing citizens to rightfully claim their ability to find the most suitable and, above all, affordable, care for them.

## 2 Background

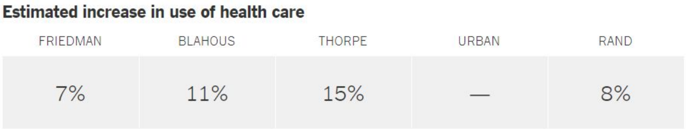
### 2.1 Addressing a social need

Unlike the vast majority of developed countries around the world, the United States does not offer a universal healthcare system for its citizens[1], whether it is paid for solely by its government or funded by taxation. This sets it apart from the rest of the developed world in a substantial way and means that most US citizens must acquire their own form of health insurance in order to pay for any medical treatment they require or face paying a considerably large amount in fees for said treatment. Indeed, the average cost of a 3-day stay in hospital costs $30,000[2] and, especially to the uninsured individual, such figures make for very grim reading.

The intent of *CompareTheHealthcare.com* is to provide an online price comparison portal for treatment and healthcare available to American citizens, with mind paid towards a citizen's location and budget. This would allow for a citizen to find treatment options available to them across the United States, emphasising the closest and most affordable options. This would address a need in American society: a need in which many working-class Americans struggle to pay for the medical treatment that they need and may not be aware of all the options available to them. For example, local healthcare providers to a particular citizen may offer care at too high a price and the portal could recommend a cheaper procedure for them in the next town over. Perhaps the procedure offered locally has had fewer discharges than those of the hospital in the next town over. All of these needs can be addressed by *CompareTheHealthcare.com* and its intuitive web portal, thus ensuring a social and ethical right for all consumers regardless of status and location to affordable healthcare that is suitable for them.

## 2.2 Promoting Health Insurance

*CompareTheHealthcare.com* highlights the difference (often a large one) between the prices paid for the same treatment by a citizen insured by Medicare and a citizen without any form of medical insurance. According to our data, differences in prices can be substantial; in the order of tens of thousands of dollars, especially with more expensive forms of treatment. Estimated prices of treatment, both insured with Medicare and uninsured, are displayed side by side to the end user, highlighting the enormous worth that is the government-sanctioned public health insurance scheme in the lives of 59.9 million people insured with Medicare in 2018[3], as well as health insurance in general for the 91.2% of the population with health insurance in 2016[4]. However, with an estimated 30 million people remaining uninsured, this represents a big gap in the market that needs to be addressed: that an estimated 30 million people are paying more than they need to for the treatment they need. A save in expenses, granted; until the day comes when they unexpectedly find themselves in the hospital with a nurse handing them a bill for far more than they bargained. *CompareTheHealthcare.com* emphasises the importance of health insurance for American citizens and why remaining uninsured is a risk not worth taking. Indeed, offering Medicare to all United States citizens over the age of 65 is one of the champion electoral promises of presidential candidate Bernie Sanders[6], with strong support amongst his followers. A Medicare For All solution as proposed by Bernie Sanders would offer insurance to an estimated 28 million more Americans[8], highly encouraging more people to seek help with paying for medical bills.



**Estimated increase in use of health care**

| FRIEDMAN | BLAHOUS | THORPE | URBAN | RAND |
|---|---|---|---|---|
| 7% | 11% | 15% | — | 8% |

*Figure 1. Table of estimated increases of healthcare usage by American citizens, courtesy of various labelled think tanks. Table courtesy of the New York Times.*

On the flip side, hospitals across the US lose billions from Medicare every year, because Medicare coughs up less in cover to your healthcare provider than your average private health insurer. This effect has been getting worse over time, meaning healthcare providers are losing more and more money thanks to Medicare every year. In 2018, even the most efficient hospitals made a negative 2.8% margin on coverage from Medicare, down from negative 1% in 2017[5]. Medicare covers less of your standard hospital bill than private insurance companies and the difference between that and what the procedure actually costs is usually written off entirely[7]. A Medicare For All solution would almost certainly cause closures of hospitals across the US, especially the more rural ones lacking in a population to support them, as well as an increase in prices to cover the losses, meaning insurance costs more for everyone.

# 3 Specification

## 3.1 Problem Specification

The main problem *CompareTheHealthcare.com* aims to overcome is that many millions of citizens in the United States are unaware of or unable to easily gain access to information on the most affordable healthcare available to them. Due to a competitive market, prices for medical treatment are widespread and highly variable across the United States, meaning two individuals' charge for healthcare on the same procedure, even with very similar circumstances, can vary widely. Our team identified a website as the most accessible, easy-to-use interface for the average American consumer, supported by a secure backend interface that communicates with a database of millions of records on procedures offered in American healthcare, based on a dataset provided by the client.

The *CompareTheHealthcare.com* portal is an interactive web interface designed to compare the price and location of a wide range of medical procedures and treatments offered across the entirety of the United States of America. These procedures cover the vast majority of all procedures offered in the US, ranging from minor throat problems to major surgery on the brain. The portal is aimed at lower- to middle-class American citizens with health insurance provided by the government-sanctioned Medicare program or with no health insurance whatsoever. The goal of the portal is to provide a range of prices for the end user for a recommended treatment option based on input keyword searches by the end user, prioritising lower prices offered at closer locations to that user, giving them a clear picture on the range of healthcare available to them based on symptoms they have. Improving awareness of the medical options available to the American consumer, regardless of status, location, race, family or career, is deemed the most important factor in the development of this project by the team behind *CompareTheHealthcare.com*.

### 3.2 Initial Plan of Work

The team, as a whole, began with researching the background behind the American health system and the provision of health insurance to the American consumer. We recognised the importance of gaining a deep understanding of the end consumer, the biggest social issues of the US and the problem faced by American consumers before tackling a solution. It was also important to realise why such a solution is necessary, as it would help to identify the functional requirements of the solution and allow it to better target the right people and fulfil its purpose.

Past this, the next step was to brainstorm what our solution would look like. It was decided, due to the popularity of the medium, to develop a web interface to communicate with a database containing the records of data on hospital procedures. This would separate the data from the consumer adequately and ensure data was stored in a large and secure location, sorted and filtered into the right format, queried based on what the user is searching for, with results provided in a quick and efficient manner.

Next was to identify the strengths of each team member so that tasks and frameworks could be implemented in the quickest and most orderly fashion. Fittingly, the team was equipped with a wide range of skills and was able to quite easily split up tasks based on who had expertise in frontend web development, backend application programming interface implementation and database configuration. This helped us massively in working cohesively and effectively to make quick progress and wrap up sprints with little to no remaining tasks to do. The idea was to have the database set up completely by the middle of the second sprint, with a prototype website available to show to the client by the end of the first sprint. As well as this, we wanted the API to be completed in a timely manner to prevent unnecessary time wastage if both the website and database were complete. As such, one team member was dedicated entirely to API implementation.

The team also recognised the importance of the report and documentation so spent a lot of time focusing on documenting the work we did to ensure our design and implementation process was audited. This was done in both written form and as comments and READMEs available in the version control repositories of the project. This helped to improve the software development lifecycle of the project as a whole as well as future maintenance of the project. The end goal was to finish the entire project with a cohesive, easy-to-use frontend, secure backend API to direct requests through and a quick and efficient database system to query and easily add data to. This included heavy documentation, as well as a user manual, on how to use the software and where it is located on version control storage for future developers to work on.

## 4 Design

An agile methodology was employed for this project where we worked in 3 sprints of 1 week each, undertaking daily stand-ups and weekly sprint retrospectives. Due to the size of the project, we thought this would be the best approach for our workflow. Therefore, requirements, user interface designs, user stories and use cases were all written prior to starting with the project. This helped us greatly in formulating our approach and gave us a clear goal of what we were looking to achieve.

### 4.1 Goals and Research

The team collectively drafted a list of goals we wanted to achieve, with speed and accuracy of user search being our most important. Crucial to achieving this was implementing an efficient design of the database, with tables for sorted and related data that belonged together, appropriate usage of foreign keys to draw relationships between data and the most efficient queries for retrieving results quickly. See Appendix B for the design of our database implementation.

Our discussions on how best to search the data revolved around analysis of the dataset we were given. Researching each field in the dataset, such as average Medicare payments for a procedure, average covered charges and total number of people discharged was important in understanding which data are most useful to the end user. Due to the lack of experience the team had with the US healthcare system, we wanted to make sure everyone was aware of the vision we had in mind for the project and what we wanted it to look like. This would help galvanise the team in their efforts and reduce the amount of confusion or disillusion any team members may have.

## 4.2 Dataset Analysis & Keyword Mapping

It was important to get an understanding of the dataset we were working with so that the right information could be given to the end user. It was quickly discovered that data was lacking: notably the inclusion of meaningful information for the end user relating to each procedure. Medical procedures and treatment were grouped into categories which were given an obscure, sometimes unhelpful name, such as "INTRACRANIAL VASCULAR PROCEDURES W PDX HEMORRHAGE W MCC". The end user, no matter how advanced in medicine they may be, is unlikely to search our web portal using those keywords. As such, the decision was taken to map each procedure name to keywords the consumer is more likely to search for. For example, the keyword "cancer" would show all procedures relating to cancer, "brain" would show procedures on the head and "ear" would map to procedures for general procedures on the ear, nose or mouth, thus allowing the user to enter these search terms on the web portal and be redirected to the group of procedures they need and providers that offer them.

For this reason, and also due to the size and complexity of the dataset, it was decided to dedicate a member of the team to the parsing of the dataset and the generation of a mapping of keywords to DRG procedures. With this map in place, it would then be linked to the frontend searching function such that a search for a keyword in the map would trigger a database query for the procedure that keyword maps to, compared to the user having to search for a procedure directly by name. This would open up the website to the end user in a much more accessible way than before and removes the need for them to memorise DRG codes for procedures and their respective definitions.

The way the team decided to go about this was to rely on synonyms for nouns in each of the DRG definitions and use these as keywords that the user can enter for which they can map to. A user is much more likely to search for "scrape" than "SKIN GRAFT EXC FOR SKIN ULCER OR CELLULITIS W/O CC/MCC". Thus, the expanse of vocabulary through which the user can query the data is greatly expanded.

## 4.3 Geolocations

With a map of hospitals present in search results being one of the functional requirements of the project, the team spent much time researching how best to provide relevant results to the user based on their location and the location of the hospitals. Unfortunately, neither of these were readily available to access, besides the address of the hospitals. Finding out where the user was was easy: most modern-day browsers have this ability built-in, using the user's computer's IP address and an online database of mappings of MAC addresses to physical location, supported by the Global Positioning System. However, beyond the address, we had no easy way of geographically pinpointing a hospital on a map.

After some research, the team came across an external dataset that mapped over 43,000 US zip codes to geographic coordinates, provided by OpenDatasoft[9]. The team ran a test query on the data to see if any zip codes already present in the hospital dataset were not present in the zip code dataset and were pleased to find that they were all present. Immediately, the problem was solved and the dataset was imported straight into our database for use with the Google Maps API.

Unfortunately, due to the size of the zip code dataset, the team was unable to determine if every US zip code was present in the dataset. The design accommodates an admin to insert a new hospital into the database but only allows for a zip code to be entered if it exists in the database already. Despite this problem, the team deemed that 43,000 unique zip codes was enough to cover this and that the likelihood of a zip code being entered that wasn't present in the database was very slim.

## 4.4 Security Considerations

Due to the sensitivity of the data and the importance of the financial figures remaining intact and accurate, the team recognised the importance of preventing unauthorised access to the level of control in which data could be deleted or, worse, modified. Because of this, the implementation of an API was very important to the team for ensuring only certain queries could be run on the database. The frontend was designed to require search terms to resolve to a specific procedure name or state name and at that point, even if malicious SQL queries were to have made it through the frontend, the API would clean the data it received and use prepared statements to be fully certain that all accesses to the database were legitimate and in good faith.

Access to administrator permissions for adding or amending data were decided to be restricted to an admin login at the frontend. The passwords for admins were to be hashed using the *bcrypt* hashing algorithm, which is believed to be secure at the time of writing. The *bcrypt* algorithm is particularly secure due to the fact that it is deliberately memory intensive. This limits the rate at which hashes can be brute-forced attacked by a malicious party. The password hashes were to be stored in the database instead of a plaintext password, thus protecting the logins if the database were to be compromised. When logging in, the API returns a password hash for a requested username, meaning even if the API can't see any plaintext passwords. Beyond this, even if an attacker were to query the API with any string for a username requesting its associated password, a *bcrypt* hash would be returned regardless. Thus, only the *bcrypt* password verification function would be able to tell if this matched with any username.

# 5 Implementation and Testing

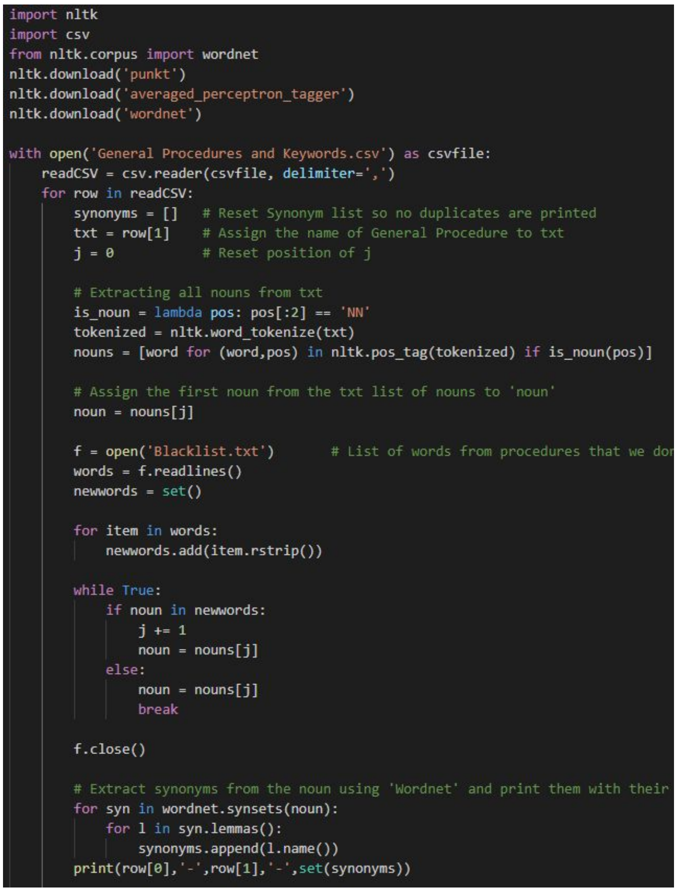
With the team split into four distinct groups, one for frontend web development, one for database implementation, one for backend API development and one for dataset parsing, progress in putting the pieces together was swift and efficient. Using the designs of how the database would store and send data in response to queries was vital and formed the backbone of development in the backend.

## 5.1 Dataset

### 5.1.1 Approach

The dataset section of the team developed a Python script to communicate with an online thesaurus database called WordNet and forward it nouns from the given DRG procedure and have it return synonyms to store in the map. This allowed for keywords to be found much more quickly compared to the manual input of synonyms for each procedure, which would have been a time-consuming process with 633 different procedures to map keywords to.

The map was hooked up to the website searching feature to ensure the smoothest and quickest retrieval of data compared to having to store keywords for procedures in the database. This followed one of the team's primary goals, which was to provide a fluid and responsive searching feature to the end user, ensuring the information they needed could be found quickly and efficiently.



```python
import nltk
import csv
from nltk.corpus import wordnet
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

with open('General Procedures and Keywords.csv') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
        synonyms = []   # Reset Synonym list so no duplicates are printed
        txt = row[1]    # Assign the name of General Procedure to txt
        j = 0           # Reset position of j

        # Extracting all nouns from txt
        is_noun = lambda pos: pos[:2] == 'NN'
        tokenized = nltk.word_tokenize(txt)
        nouns = [word for (word,pos) in nltk.pos_tag(tokenized) if is_noun(pos)]

        # Assign the first noun from the txt list of nouns to 'noun'
        noun = nouns[j]

        f = open('Blacklist.txt')       # List of words from procedures that we dor
        words = f.readlines()
        newwords = set()

        for item in words:
            newwords.add(item.rstrip())

        while True:
            if noun in newwords:
                j += 1
                noun = nouns[j]
            else:
                noun = nouns[j]
                break

        f.close()

        # Extract synonyms from the noun using 'Wordnet' and print them with their
        for syn in wordnet.synsets(noun):
            for l in syn.lemmas():
                synonyms.append(l.name())
        print(row[0],'-',row[1],'-',set(synonyms))
```

*Figure 2. Excerpt of Python script for fetching synonyms for a given keyword, used for the web search. Note the use of an external thesaurus dataset called* nltk

### 5.1.2 Problems

However, the technique to generate synonyms for given nouns from the DRG definitions was not always successful, as some entirely unrelated or unsuitable words could be returned from it for sometimes sensitive DRG procedure groups, especially with the likes of "ANAL AND STOMAL PROCEDURES", and "VAGINAL DELIVERY". As such, some inappropriate keywords had to be filtered out and replaced with more appropriate terminology. The keywords that were not allowed were stored in a blacklist text file that could be easily amended if new blacklisted keywords were thought of. Overall, the use of this script helped a lot in reducing the time it took to carry out this function, despite its flaws.

As well as this, the dataset itself was partially corrupted, likely because it was an amalgamation of several years' worth of data from different hospitals, much of which is in the same format but, when combined, clashes. The team discovered issues with data for 2016/17, which was formatted differently from the rest that caused problems when converting the JSON dataset to a comma-separated values format. These problems included commas suddenly appearing in numerical values, which conflicted with the very nature of how CSV files are formatted, as well as dollar signs suddenly appearing in data which SQL was not partial to. When these problems manifested, it often required a complete re-import of the data into the database, which could take hours.

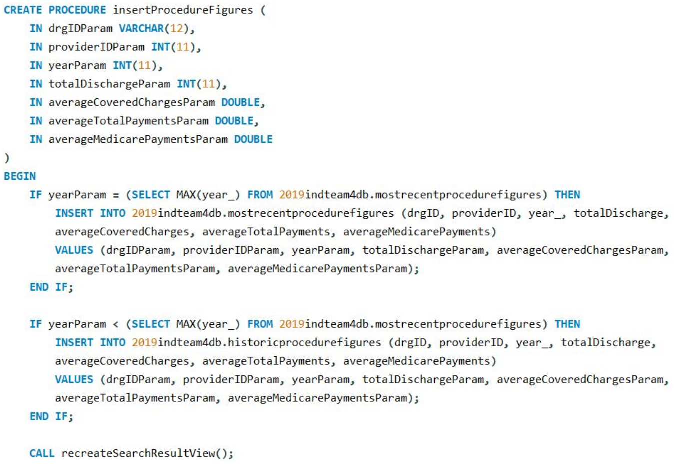
With this in mind, it was more important than ever for the team to be efficient in their work and design solutions to work around the problems encountered and mock or stub data we could not rely on. For example, while the database or API was down and being fixed, the frontend team relied on a dummy data file to check their results were being formatted properly and map pointers were being correctly displayed. This was planned for and assumed to occur so that the team was prepared for all possible scenarios.

## 5.2 Database

### 5.2.1 Approach

The database section of the team cleaned the dataset of procedures at American hospitals by importing it into a CSV file. This proved to take longer than expected, partly because of issues in converting the JSON data to comma-separated value data and how the SQL database struggled to properly parse this, leading to null and void values in several places in the database that required another attempted import. The size of the dataset made the importing and the validation of the data more difficult, meaning outliers in the data that shouldn't be present were only discovered upon querying the data in several different ways. The solution to this, as well as the overall slow performance of the database at times due to other teams doing the same thing, was to implement a local copy of the database using LAMP stack technology before checking it and publishing it to the online Silva database system. This improved the performance of the database but only allowed one user to interact with the new data while it was on their local machine. Thankfully, while that was happening, these team members could design and write up the necessary queries and stored procedures for the database and forward them to the API section of the team. This helped to improve the implementation time of the API in turn.

Another solution to improve the performance of the database and to separate data into meaningful places was to have one table for the most recent data per procedure per hospital and another for the historical data of the same procedures of the same hospitals. This massively improved performance due to the large amount of less relevant data being moved to a table less queried. It was also configured so that new insertions of data would join the right table of data, e.g. if a previous year of data were to be added, it would go into the historical table, otherwise it would join the most recent data table.



```
CREATE PROCEDURE insertProcedureFigures (
    IN drgIDParam VARCHAR(12),
    IN providerIDParam INT(11),
    IN yearParam INT(11),
    IN totalDischargeParam INT(11),
    IN averageCoveredChargesParam DOUBLE,
    IN averageTotalPaymentsParam DOUBLE,
    IN averageMedicarePaymentsParam DOUBLE
)
BEGIN
    IF yearParam = (SELECT MAX(year_) FROM 2019indteam4db.mostrecentprocedurefigures) THEN
        INSERT INTO 2019indteam4db.mostrecentprocedurefigures (drgID, providerID, year_, totalDischarge,
        averageCoveredCharges, averageTotalPayments, averageMedicarePayments)
        VALUES (drgIDParam, providerIDParam, yearParam, totalDischargeParam, averageCoveredChargesParam,
        averageTotalPaymentsParam, averageMedicarePaymentsParam);
    END IF;

    IF yearParam < (SELECT MAX(year_) FROM 2019indteam4db.mostrecentprocedurefigures) THEN
        INSERT INTO 2019indteam4db.historicprocedurefigures (drgID, providerID, year_, totalDischarge,
        averageCoveredCharges, averageTotalPayments, averageMedicarePayments)
        VALUES (drgIDParam, providerIDParam, yearParam, totalDischargeParam, averageCoveredChargesParam,
        averageTotalPaymentsParam, averageMedicarePaymentsParam);
    END IF;

    CALL recreateSearchResultView();
```

*Figure 3. Query for inserting figures on a given procedure for a given provider. Note the IF statement, comparing the given year against the most recent data for that procedure at that hospital and adding the new data to the appropriate table.*

### 5.2.2 Problems

Unfortunately, implementation of the database was hampered throughout the project by many different bugs, some of which the team could not have anticipated. One of the most frustrating of these was to do with string collation conflicts on field comparison queries, thus grinding testing of the API and frontend to a halt. What made this worse was that copies of the database running on local machines did not seem to suffer from this problem, so it was impossible to predict if the issue remained once the database was amended and reuploaded, thus slowing down development even more, not just on the database but the API and frontend trying to retrieve results and having no luck. The team as a whole worked hard to overcome this by doing lots of research into the issue and, through trial and error of different character sets, collated the data as a whole to fit one schema, including any new imports of datasets.

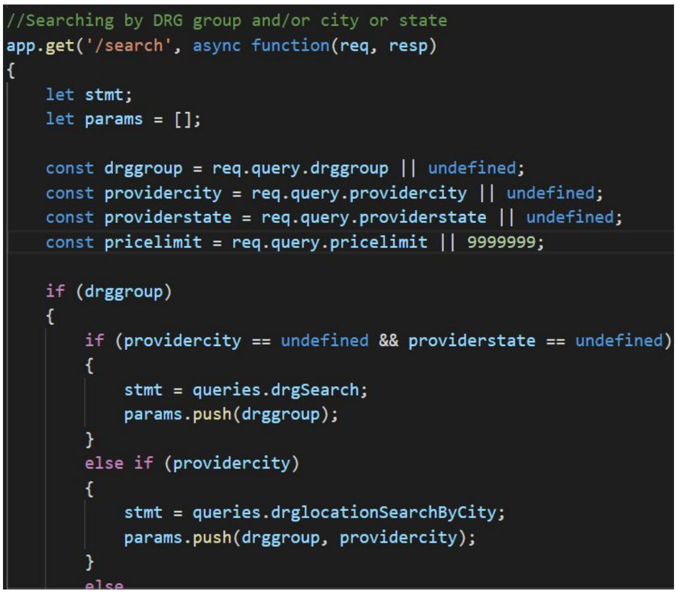
Further to this, with many other teams also relying on University-provided databases hosted on the same server, queries and changes of the database often took a long time due to network bandwidth limitations and poor server-handling. This could often result in queries taking far longer than they should, time out or not execute at all. Frontend testing of searches was hampered by this as it was difficult to diagnose the problem and learn if the issue was present in the frontend, the API or the database. The use of a local database for uploading and changing lots of data for upload to the main database later while the rest of the team made smaller queries on the main database helped here, as it reduced the load on the server and helped all teams work a bit faster.

## 5.3 Application Program Interface

### 5.3.1 Approach

A Javascript API was developed solely to interact with and forward results from the database. This decision was taken for a number of reasons:

1. Interacting with the database was to be a highly controlled operation, executed by those with authorisation only
2. The database and the security of the database could be more easily managed by separating them away from the frontend web development, which should be focusing more on the user experience than the interactions with the database
3. The team's prior experiences with using PHP to interact with the database were not enjoyable and the team did not believe that mixing SQL queries written in PHP with the frontend code was a particularly good idea.



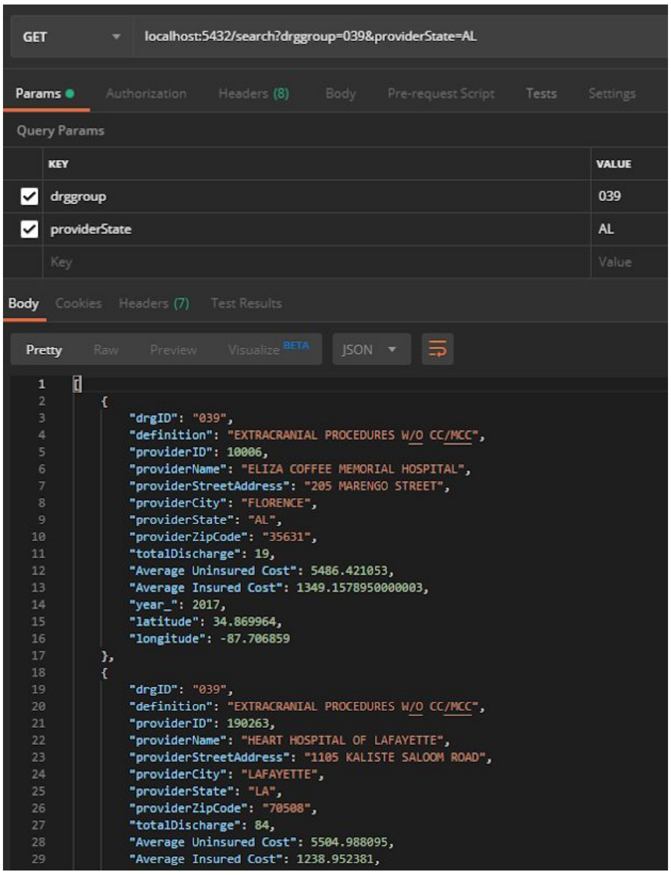
```
//Searching by DRG group and/or city or state
app.get('/search', async function(req, resp)
{
    let stmt;
    let params = [];

    const drggroup = req.query.drggroup || undefined;
    const providercity = req.query.providercity || undefined;
    const providerstate = req.query.providerstate || undefined;
    const pricelimit = req.query.pricelimit || 9999999;

    if (drggroup)
    {
        if (providercity == undefined && providerstate == undefined)
        {
            stmt = queries.drgSearch;
            params.push(drggroup);
        }
        else if (providercity)
        {
            stmt = queries.drglocationSearchByCity;
            params.push(drggroup, providercity);
        }
        else
```

*Figure 4. Excerpt of code for the search route in the API, through which the main user search is performed. User can call a search using just a DRG definition, or city or state, with an optional procedure price limit.*

The development of the API was straightforward and had little to no technical hitches. Team members working on this had prior experience with developing web and RESTful APIs in the past so the team was readily equipped with someone who had the necessary skills and expertise to create a secure interface between website and database. Thus, development was swift and any changes that needed to be made could be done in a timely manner. Different web routes were programmed to cater for different queries, with one for inserting new records and one for retrieving all data for admin users only as well as a universal procedure search available for all users, taking search parameters by procedure, city or state. Data was also kept secure by using POST compared to GET requests. As well as this, only specific stored procedures can be executed, as defined in a config file in the API, meaning an end user will be unable to construct their own queries that could exploit the database to return unauthorised data.



localhost:5432/search?drggroup=039&providerState=AL

|  | KEY | VALUE |
|---|---|---|
| ☑ | drggroup | 039 |
| ☑ | providerState | AL |
|  | Key | Value |

```
[
    {
        "drgID": "039",
        "definition": "EXTRACRANIAL PROCEDURES W/O CC/MCC",
        "providerID": 10006,
        "providerName": "ELIZA COFFEE MEMORIAL HOSPITAL",
        "providerStreetAddress": "205 MARENGO STREET",
        "providerCity": "FLORENCE",
        "providerState": "AL",
        "providerZipCode": "35631",
        "totalDischarge": 19,
        "Average Uninsured Cost": 5486.421053,
        "Average Insured Cost": 1349.1578950000003,
        "year_": 2017,
        "latitude": 34.869964,
        "longitude": -87.706859
    },
    {
        "drgID": "039",
        "definition": "EXTRACRANIAL PROCEDURES W/O CC/MCC",
        "providerID": 190263,
        "providerName": "HEART HOSPITAL OF LAFAYETTE",
        "providerStreetAddress": "1105 KALISTE SALOOM ROAD",
        "providerCity": "LAFAYETTE",
        "providerState": "LA",
        "providerZipCode": "70508",
        "totalDischarge": 84,
        "Average Uninsured Cost": 5504.988095,
        "Average Insured Cost": 1238.952381,
```

*Figure 5. Use of Postman API client to test the Javascript API. This request was for a search for "Extracranial procedures" and shows a preview of the real data returned to the frontend from the API.*

### 5.3.2 Problems

Testing of the API proved a little more challenging; due to only holding experience with testing Flask APIs, the team had to research the best techniques and frameworks for testing both synchronous and asynchronous functions. The result was Mocha: a highly adaptive and flexible Javascript testing library that allowed for several unit tests to be developed, thus further ensuring the security of the API and, by extension, the database.

A lot of time was spent researching how to test API routes that required a POST request compared to a GET request.

Including parameters in the test query which could be extracted by the receiving function proved very challenging, despite many apparent solutions online that didn't seem to suffer from the same issue. This meant some unit and integration tests could not be carried out and manual HTTP testing was performed through the Postman API client. The use of stored procedures helped this, as it only required the stored procedures to be tested out to ensure they were rigorous, then they could simply be called by the API with no problems expected.

## 5.4 Web Frontend

### 5.4.1 Approach

The web team was responsible for the final and most important product: what the end user sees on their screens. The decision to not rely on a full development framework, such as Angular or React, in the development of the team's frontend proved to be a good one. Bootstrap provided a nice template for which to base our design on and the lack of a framework allowed for the flexibility to mesh different technologies and languages, such as Javascript, PHP and Python, together with ease; something that a web framework may have restricted a little. As well as this, the web team's combined experience in one specific web framework did not allow for one to be easily selected, as significant time may have been spent learning and understanding the new framework. Although the team recognised the potential benefits of using an established framework, they were not prepared to run the risk of the learning curve impacting the quality of the product. As such, the web team could draw on previous experience with native web tools and scripting languages to collaborate on the frontend effectively.

Within the first few days of the initial sprint, most of the page layouts were complete. This was mainly due to the straightforward components provided by the Bootstrap framework. From then on, the next step was to make a link to the backend to facilitate the exchange of data from our database. Our API thankfully made this very simple. We were able to simply construct an XML request object that would connect to one of our endpoints. Thus, production data could then be used on the frontend.

One of the client's requirements was to be able to sort search results in a number of ways. We thought that a good way to approach this would be to use a table with Javascript functions attached. After some research, the web team settled on using a plugin called Bootstrap Table[10]. This plugin allowed a number of attributes to be added to a standard table which would enable additional functions such as sorting by column and pagination. We experimented with the performance of server-side pagination as well as client-side. We understood that server-side would put less strain on the user's browser and so we tested it. Our results showed that the client-side was able to handle the number of results responsively. The decision was made to stick with client-side pagination as the trade-off in responsiveness offered by server-side would hurt the user experience.

To visualise the results of a search, the team employed the use of the Google Maps API. As we wanted to keep our expenses to a minimum, it was important to limit the calls made to the Google API. When we run the distance calculation on our results, we work out the straight line distance rather than the driving route distance offered by Google, thus saving on calculation time. If there was the budget for accurate route distances then our team would have implemented this.

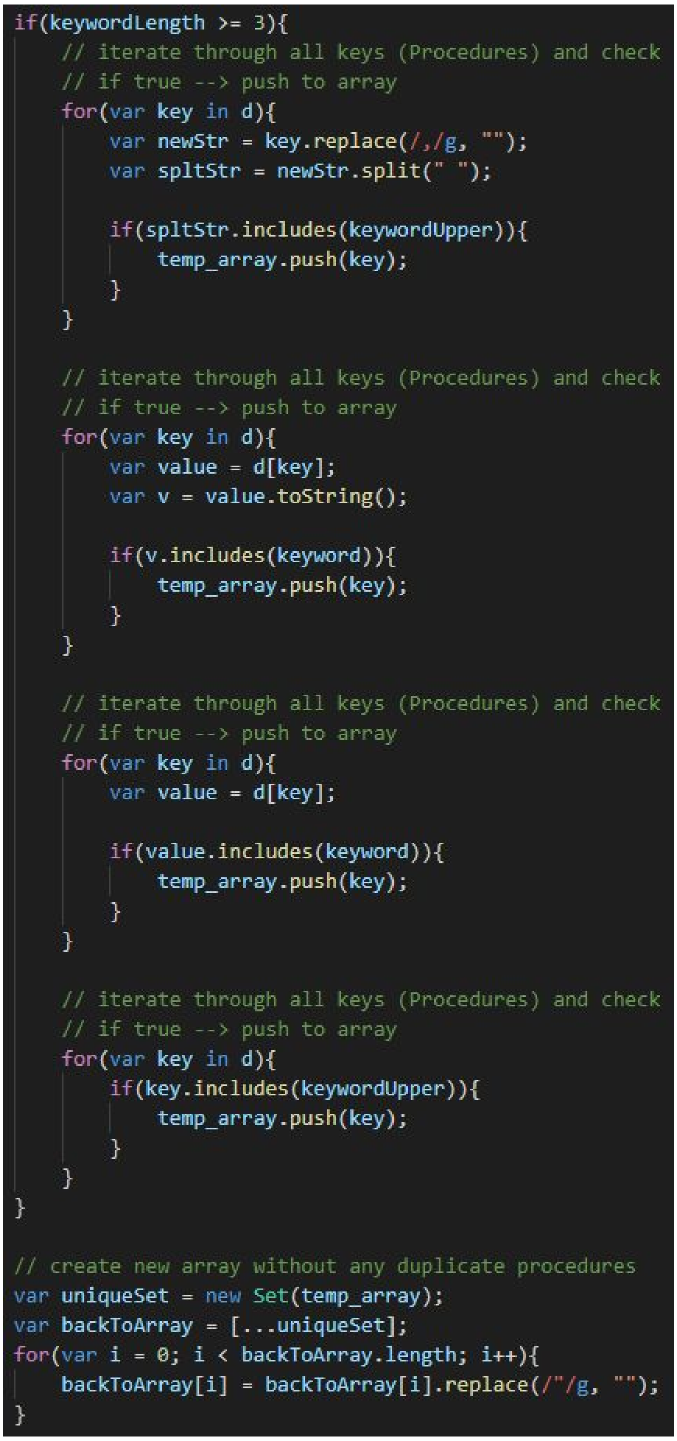
The client liked our idea of utilizing historic procedure data on a detail view. We wanted to plot this data on a stepped graph and, after looking at various charting plugins, we noticed that many were free for students but not if working on behalf of a third-party client. Since this was our situation, we eventually found our way to the Google Charts library. The web team found Google Charts easy to use but somewhat limiting in some areas. However, it provided the features that we required and so we went ahead. We also wanted to try and predict the future cost of a procedure on these charts based on previous data. The team were well aware that a full machine-learning solution was not realistic in our situation. Instead, due to deadlines, we opted for a more basic prediction based on the average difference in cost in previous years.

### 5.4.2 Problems

After the initial iteration of the results table we soon realised that this didn't include the distance data which was still to be calculated. The solution that came to us was to fetch the results from our API first, parse them into JSON, iterate through each and add a new calculated distance key to each. After this the table was then initialised with the amended data and sorted by distance by default.

Another significant problem was working out how to dynamically link the results of the table to markers on the Google map. After quite some time looking into documentation, we found that our table plugin supported a number of custom events, one of which was triggered every time the table was rendered, i.e. every time it was sorted, searched or traversed. This allowed the team to fetch the data of the newly rendered table page and update the map markers accordingly.

Asynchronous Javascript functions were a fairly new concept to some of the team members. Many of the Google API functions were asynchronous by default. It was therefore important to learn how to work with them. A common issue experienced with these was any variable inside one of these functions couldn't be exported into a global scope. One solution was to make a function synchronous when possible. However, this is often considered bad practice as it forces the user to wait on the function to resolve. In some cases, the variable could be stored in the session for use on another page. Finally, we had the option to group our initialisation functions together before the page rendered. In the end, we used a combination of these to achieve the desired outcome. If the duration of the project was longer, more time would have been invested in restructuring and refactoring the Javascript code involved.



```javascript
if(keywordLength >= 3){
    // iterate through all keys (Procedures) and check
    // if true --> push to array
    for(var key in d){
        var newStr = key.replace(/,/g, "");
        var spltStr = newStr.split(" ");

        if(spltStr.includes(keywordUpper)){
            temp_array.push(key);
        }
    }

    // iterate through all keys (Procedures) and check
    // if true --> push to array
    for(var key in d){
        var value = d[key];
        var v = value.toString();

        if(v.includes(keyword)){
            temp_array.push(key);
        }
    }

    // iterate through all keys (Procedures) and check
    // if true --> push to array
    for(var key in d){
        var value = d[key];

        if(value.includes(keyword)){
            temp_array.push(key);
        }
    }

    // iterate through all keys (Procedures) and check
    // if true --> push to array
    for(var key in d){
        if(key.includes(keywordUpper)){
            temp_array.push(key);
        }
    }
}

// create new array without any duplicate procedures
var uniqueSet = new Set(temp_array);
var backToArray = [...uniqueSet];
for(var i = 0; i < backToArray.length; i++){
    backToArray[i] = backToArray[i].replace(/"/g, "");
}
```

*Figure 6. Excerpt of Javascript code to match a given keyword to an explicit DRG definition so that a search can be called.*

## 5.5 Performance

### 5.5.1 Queries

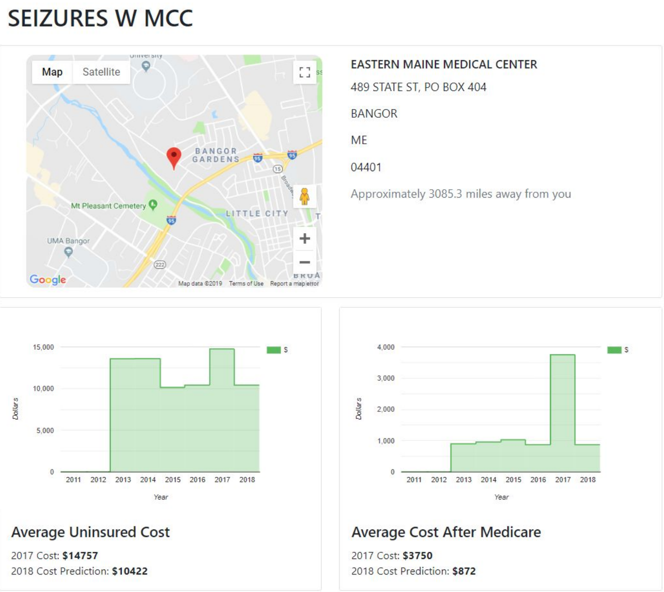
For a while during our implementation, our query performance times were lacking when compared to other teams and their individual databases. This wasn’t something that could be explained by the type of database being used (SQL, Mongo, MSSQL, etc.) nor by the amount of data we were working with (all teams were using the same dataset) nor by the structure of our tables (tables for DRG procedures, hospitals, yearly figures respectively). Despite reasonably efficient queries, response times for more general searches were typically upwards of 40 seconds: an unacceptable length of time for a production user. While all of this was going on, as well as the other problems we were experiencing with the database such as the collation issues, other teams were possessing remarkably quick query times.

The database team worked full-on researching this issue and tested several different techniques such as the type of joins used in the queries and the exact keywords used and finally a solution was found when specific views were implemented. Views were made specifically for search results, the providers, zip code coordinates and financial figures for procedures at hospitals for most recent and historical data. Query execution times were immediately sliced by over 98%, with all searches taking no longer than a second from this point onwards. This proved to be a large boost to the project and massively improved our overall performance in achieving the goals we set out to achieve in a responsive and efficient search to the end user. The client seemed very impressed with the progress that was made in this regard (see Appendix A).

### 5.5.2 Search Results

The search results for a user's search query were displayed in a paginated Javascript table, showing the name of the procedure, the hospital offering it, its address, distance from the user's location, the average uninsured cost of the procedure at that hospital, the average Medicare-insured cost of the procedure at that hospital and the year for which these figures were last gathered. The results are sorted into pages of ten results each. This gives the relevant data on the procedure back to the end user, with the option to click on a result to give further details on the offered procedure to the user. The paginated table also allows for filtering of the received data, along with a search bar for filtering hospital names by a user-input search term.

The API returns all results to a query in one go, which is then filtered and limited by the paginated table so as to not overwhelm the user and highlight the most relevant results. Artificial pagination was tested server-side using query limits but this concluded that pagination worked better client-side. Although search queries fetched results faster if artificial pagination with limits was specified, this also culminated in more queries, as another query would need to be called for the next set of results if the user switched between pages, using unnecessary amounts of bandwidth. Beyond this, thanks to the optimisations made in query running times, all of the search results could be returned in a small amount of time anyway so the team concluded that it wasn't necessary to use limits in the procedures called on the database.



SEIZURES W MCC

EASTERN MAINE MEDICAL CENTER
489 STATE ST, PO BOX 404
BANGOR
ME
04401
Approximately 3085.3 miles away from you

Average Uninsured Cost
2017 Cost: **$14757**
2018 Cost Prediction: **$10422**

Average Cost After Medicare
2017 Cost: **$3750**
2018 Cost Prediction: **$872**

*Figure 7. Preview of the details screen on the web portal, showing further information on a particular procedure at a particular hospital. Note the graphs at the bottom showing how the price of the procedure has changed over time.*

Included in the details page are:

- A more detailed map of the location of the hospital
- A graph of how the price of that procedure has changed over every year that procedure was offered at that hospital, both for the Medicare-insured and uninsured individual
- Full hospital address
- Medicare-insured and uninsured price
- Distance (as the crow flies) from the end user's given location

### 5.5.3 Mapping of hospitals

Implementing a map of hospitals offering procedure search results proved tricky for a while. Based on research of techniques used by other groups, mapping all search results in one go would likely prove difficult, both in computing power and time required. The more points required to be displayed, the more memory this used and the more the website slowed down. The time to compute the pointers would, therefore take longer but, beyond this, vague searches returning thousands of results would be less useful to the end user because the map is completely covered with pointers to the point where other useful data from the map, such as directional routes, were difficult, if not impossible, to see.

As such, the decision was taken to limit the number of pointers to the ten results currently displayed to the user. This matches the pointers to what the user is seeing on-screen in the search results. By clicking on a different page number, the pointers were updated with the next set of results. This helped performance a lot and actually took a negligible amount of time to run even compared to the main search, which could easily take less than a second.

### 5.6 Administrative Login

Full implementation of the ability for an admin to login and perform insertion and updating tasks on the data was tackled in the third sprint. The login section of the website uses a PHP prepared statement to extract the *bcrypt* hashed password from the database for the user-input username and performs a password verification comparison between that and the user-input password. Once logged in, an admin has the ability to:

- Insert a DRG definition/procedure
- Insert a hospital and its address
- Insert financial figures for a procedure offered at a hospital for a given year
- Update financial figures for a procedure offered at a hospital for a given year
- Update hospital details

These functions are offered to the admin through forms in which they can input the necessary data. Once information is published through the form, it is available for the end user to browse, allowing for the most up-to-date data to be continuously provided to consumers.

## 6 Evaluation

The team carried out eight individual user evaluations on the web portal with a mix of external users of varying backgrounds, genders and experience in computing. This allowed us to get a high range of feedback coverage, ensuring that users of all skill levels were able to use the site. It also allowed us to spot more problems with the project as the range of users allowed for the website to be used and approached in differing ways, allowing for errors we as a team may not have discovered to be found.

Users were asked to read a participant information sheet describing the university's policy on seeking ethical approval from research participants, how their feedback is stored and the purpose of the study and their involvement. To this end, participants also agreed to and signed an informed consent form that gave their written approval for participating in the study.

A questionnaire was prepared for users to fill out during the evaluation. It included sections for general questions, such as their thoughts on the website design and usability to scenario-based questions where the user poses as an American citizen trying to use the site in a specific context. One scenario was specific, with clear information on which fields to search on and the other scenario is deliberately vague with no help given from the researcher, allowing the website to be tested in an isolated environment that mimics how the website would be used in production. The user's performance in this specific category is highly important towards understanding how accessible and usable our web portal is.

The most promising feedback the team received in the evaluations was the accessibility and ease of usability of the site. Everybody surveyed believed the website to be very easy and intuitive to use, along with a simple and appealing design. Most surveyees also believed the portal provided the right amount of information and never provided too much to the end user.

However, the results of the survey indicate the information given to and required of the end user in the search is too high-level. A surveyee studying medicine attempted lots of complex searches with very low-level and specialised vocabulary, almost none of which the web portal was able to find or resolve to a valid search term. The surveyee (User C) expressed disappointment in the lack of details and descriptions of the procedures being offered and also concern that end users weren't knowing what they could be signing up to. Other surveyees also seemed confused with the procedure names and search terms, asking for further clarification when making a search about how to browse the data.

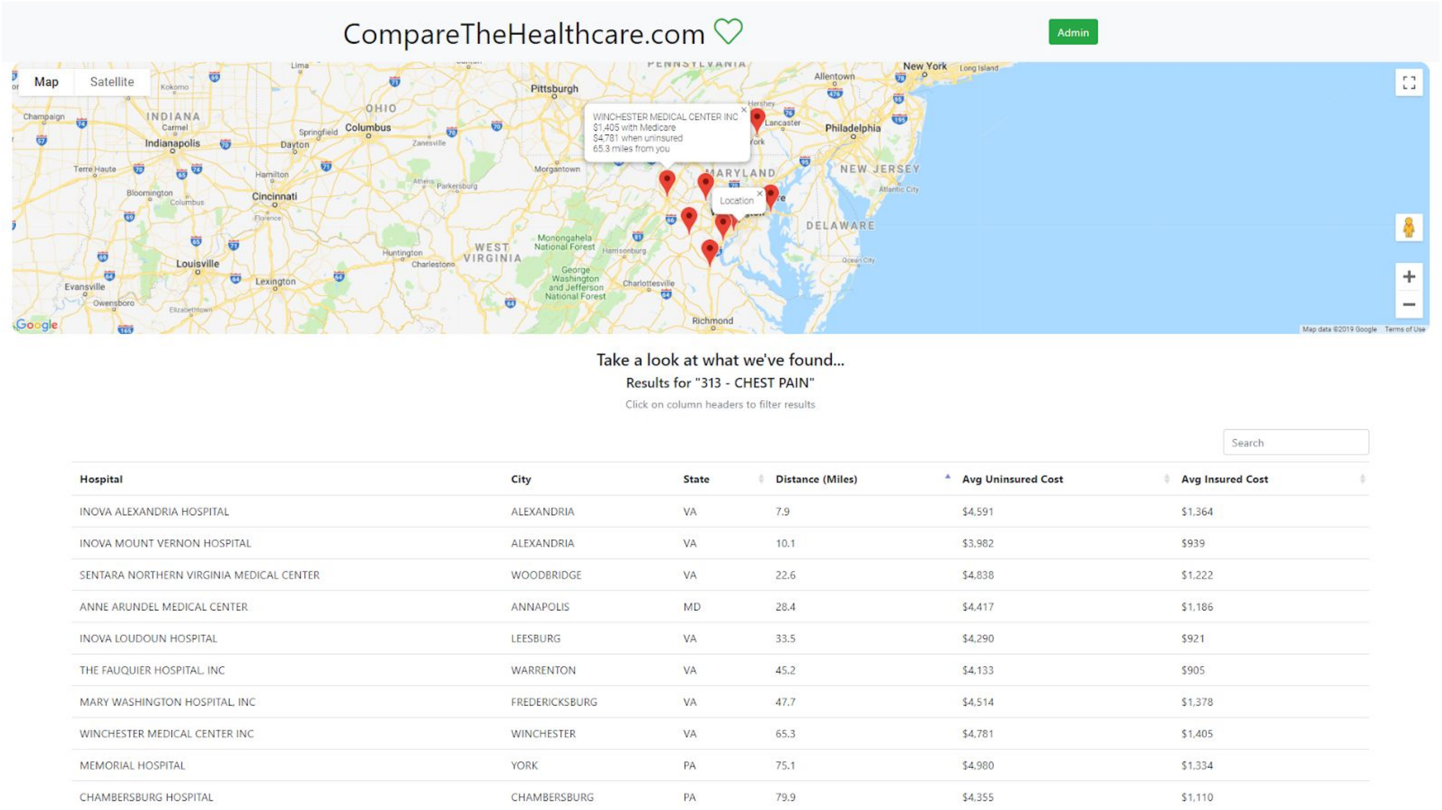
The team took the feedback received from the evaluations on board. The concerns raised by User C were highly important in addressing the issue of an incomplete dataset: there simply isn't enough information on the procedures being offered to the end user. This results in obvious confusion and disparity between the end user and the jargon they are receiving. The team also acknowledged that the search could have been improved; despite our best efforts to map more high-level dialogue to specific procedures, inevitably some vocabulary will be skipped. Notes were taken of what the surveyees were searching for during the evaluations and these search terms were added to the keyword map in an effort to ensure the most popular search terms are represented.

The full, anonymised results of the user evaluations, as well as accompanying charts and statistics, can be found in Appendix G.

## 7 Description of the final product

The final version of *CompareTheHealthcare.com* is a responsive, accessible and refined portal with an intuitive design for all users of all web experiences. The homepage is bright and decluttered, with proportional space for all sections, emphasised on the headline search feature that the user's eye is immediately drawn to. Results are displayed for a user's search query below the search box, alongside a map displaying the location of the top 10 results relative to the user's location or a location searched for. This gives an interactive ability to the end user, along with a fluid design that displays information relevant to the user in a formal and clear manner.

Provided for admin users is the ability to log in to the website, allowing for more control over the data present on the website. Among several functions available, an admin can add hospitals and DRG procedures, as well as financial figures for a specific year at a specific hospital. Past figures can be amended if errors are found in collected JSON data and hospitals can be amended if names or locations are changed. This gives a large degree of flexibility with the data that allows for new information to be quickly added for consumers to see straight away.



WINCHESTER MEDICAL CENTER INC
$1,405 with Medicare
$4,781 when uninsured
65.3 miles from you

Take a look at what we've found...

Results for "313 - CHEST PAIN"

Click on column headers to filter results

Search

| Hospital | City | State | Distance (Miles) | Avg Uninsured Cost | Avg Insured Cost |
|---|---|---|---|---|---|
| INOVA ALEXANDRIA HOSPITAL | ALEXANDRIA | VA | 7.9 | $4,591 | $1,364 |
| INOVA MOUNT VERNON HOSPITAL | ALEXANDRIA | VA | 10.1 | $3,982 | $939 |
| SENTARA NORTHERN VIRGINIA MEDICAL CENTER | WOODBRIDGE | VA | 22.6 | $4,838 | $1,222 |
| ANNE ARUNDEL MEDICAL CENTER | ANNAPOLIS | MD | 28.4 | $4,417 | $1,186 |
| INOVA LOUDOUN HOSPITAL | LEESBURG | VA | 33.5 | $4,290 | $921 |
| THE FAUQUIER HOSPITAL, INC | WARRENTON | VA | 45.2 | $4,133 | $905 |
| MARY WASHINGTON HOSPITAL, INC | FREDERICKSBURG | VA | 47.7 | $4,514 | $1,378 |
| WINCHESTER MEDICAL CENTER INC | WINCHESTER | VA | 65.3 | $4,781 | $1,405 |
| MEMORIAL HOSPITAL | YORK | PA | 75.1 | $4,980 | $1,334 |
| CHAMBERSBURG HOSPITAL | CHAMBERSBURG | PA | 79.9 | $4,355 | $1,110 |

*Figure 8. Preview of the results screen on the web portal for a search on "chest pain". Note the results are in ascending order of distance to provider, as visually displayed by the map*

## 8 Summary and Conclusions

The team is proud to report the delivery of a solid and sophisticated project whilst applying a strict agile methodology to its work. The main goals of the project were to provide a useful end-product whilst applying an efficient and robust agile approach to our problems. The team regulated itself effectively by adhering to product and sprint backlogs, daily stand-ups, sprint planning, reviews and retrospectives on how to improve. This massively improved the team's functionality and collaboration efforts, multiplying our collective output by a significant amount.

We delivered on many of the goals we set out to achieve, especially on the desire to provide an efficient and fast search of a large amount of data; a problem that has plagued many a database admin and software developer in a world of big data. Significant progress was made in this area alone and the team massively optimised the search query times, reducing them by over 98% on average. Considering the sheer amount of problems the team encountered with the database, this was a big achievement and a problem we were happy to overcome.

We also delivered on security by relying on stored procedures and prepared variable insertion into those queries through a custom API, as well as keywords on the frontend search required to resolve to a safe procedure name. This significantly lowers the chances of our project coming under a successful attack from SQL injection. The use of hashed passwords, verified by *bcrypt*, ensures an attacker never gains access to plaintext passwords, making them unable to login to insert or amend data, thus ensuring the validity and reputability of information to the consumer.

Finally, the project was finalised in a state where it can be readily expanded or built upon. Future hospitals and procedure figures for each year can be readily added, meaning the website can grow in what it offers and continue to be relevant this year and beyond. Erroneous data can be amended to fix mistakes made and the database is prepared for future datasets to be imported and used by the consumer.

Multiple issues were encountered during the development of this project; however, the team always collaborated effectively to solve them. If asked to change our approach in the next project, we would have likely planned a little more thoroughly, either by online research or drawing on past experience, to have anticipated the problems we were going to have further down the line. For example, more thorough dataset analysis may have helped avoid the issues the team ran into when importing the dataset into the database; this wasted significant time due to long data import and query times and vague error messages reported by MySQL.

On top of this, the team sometimes found itself stretched in several different directions, which improved our addressability of multiple problems but reduced collaboration on smaller issues that could have used a helping hand. A solution to this would have been pair programming, which would have improved the overall functionality of the code we wrote and prevented more bugs; however, the team likely wouldn't have achieved so many story points and ticked off as many tasks in its backlogs as it would with this approach. Oftentimes, the most crucial part of an effective team is finding a balance, through trial and error, of working, helping to achieve the highest and most efficient production levels possible.

The team is optimistic of a successful future for *CompareTheHealthcare.com* and its members are hopeful of working together in the future again.

## Acknowledgments

The team would like to thank the project client, Craneware, and specifically its representatives Connor Haining and James Read, for their collaboration with us over the three weeks that this project was in progress. We conclude our project hoping the overall collaboration between Craneware and the University of Dundee has been a successful venture.

A special thanks go to the Computing department's resident jack-of-all-trades, Mahamadou Niakaté, for his help in setting up the team's SQL database and addressing all the issues that we and all the other team's had, especially with the constant server restarts, slow-downs and shutdowns. Finally, we would like to also thank Dr Craig Ramsay and Dr Brian Pluss for running the module.

## References

[1] Amadeo, K. (2018). Universal Health Care in Different Countries, Pros and Cons of Each. [online] The Balance. Available at: https://www.thebalance.com/universal-health-care-4156211 [Accessed 21 Sep. 2019].

[2] HealthCare.gov. (2019). Health coverage protects you from high medical costs. [online] Available at: https://www.healthcare.gov/why-coverage-is-important/protection-from-high-medical-costs/ [Accessed 21 Sep. 2019].

[3] Centers for Medicare & Medicaid Services. (2019). 2019 ANNUAL REPORT OF THE BOARDS OF TRUSTEES OF THE FEDERAL HOSPITAL INSURANCE AND FEDERAL SUPPLEMENTARY MEDICAL INSURANCE TRUST FUNDS. [online] Available at: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/ReportsTrustFunds/Downloads/TR2019.pdf [Accessed 24 Sep. 2019].

[4] Barnett, J. and Berchick, E. (2017). Income, Poverty, and Health Insurance Coverage in the United States. [online] Census.gov. Available at: https://www.census.gov/content/dam/Census/library/publications/2017/demo/p60-260.pdf [Accessed 24 Sep. 2019].

[5] Morse, S. (2019). Efficient hospitals operate on -2% margins in Medicare payments, MedPAC reports. [online] Healthcare Finance News. Available at: https://www.healthcarefinancenews.com/news/efficient-hospitals-operate-2-margins-medicare-payments-medpac-reports [Accessed 24 Sep. 2019].

[6] Sanders, B. (2016). Medicare for All. [online] Bernie Sanders - Official Campaign Website. Available at: https://berniesanders.com/issues/medicare-for-all/ [Accessed 24 Sep. 2019].

[7] Cms.gov. (2018). Inpatient/Outpatient PUFs: Frequently Asked Questions. [online] Available at: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Inpatient_Outpatient_FAQ.pdf [Accessed 24 Sep. 2019].

[8] Katz, J., Quealy, K. and Sanger-Katz, M. (2019). Would 'Medicare for All' Save Billions or Cost Billions?. [online] Nytimes.com. Available at: https://www.nytimes.com/interactive/2019/04/10/upshot/medicare-for-all-bernie-sanders-cost-estimates.html [Accessed 24 Sep. 2019].

[9] Public.opendatasoft.com. (n.d.). US Zip Code Latitude and Longitude. [online] Available at: https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/table/ [Accessed 30 Sep. 2019]

[10] Bootstrap-table.com (2019). Documentation. [online]
Available at:
https://bootstrap-table.com/docs/getting-started/introduction/
[Accessed 18 Sep. 2019].

# Appendix A

**Title**: Minutes of meetings held with the client and the management teams

**Relevance**: This document holds formal minutes, drafted by Sam Glendenning and John Parsons, of the meetings held between the team and the client and management teams. It gives intimate details of what topics were discussed, who said what and the reactions and comments made by each party.

Type: Management Team Meeting
Date: 16/09/19
Time: 15:45
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Brian Pluss, Victor Yu, Sam Glendenning, John Parsons, Youssef Aloulou, Adam Munro, Saif Fardan
Absent: None
Minutes by Sam Glendenning

**TOPIC**
**Testing familiarity with project**
CR questioned our familiarity with the project and how we were getting started with it. JP responded by talking about our extraction and parsing of the given dataset, along with construction of a web frontend and API to communicate with the database. CR and BP approved these methods.

**TOPIC**
**Questioning the ethics approval requirement**
SG asked CR about the meaning of ethical approval and its requirement. CR responded by describing the need for external moderation and testing of research-based projects and how written approval is required to conduct this testing on external participants and informing them on the usage of their data.

-----------------------------------------------------------------------------------------------------------

Type: Management Team Meeting (Team Leader)
Date: 19/09/19
Time: 11:30
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Brian Pluss, John Parsons
Absent: None
Minutes by John Parsons

**TOPIC**
**Team Structure & Progress**
CR asked JP how he felt the team was operating. JP outlined the team methodology and work routine. He expressed that the team was working well and everyone was contributing to the project goal. JP briefly summarised the reasons for choosing our particular development environment.

**TOPIC**
**Questions and Problems**
CR responded positively to the progress of the team and asked if JP had any questions or problems. JP brought up the difficulties encountered with importing the data but assured that despite this, progress was being made in other areas. Following this there was a brief discussion regarding the lack of out-of-hours access to the building. CR & BP agreed that this needed to be resolved and promised to look into the matter.

Type: Client Team Meeting
Date: 23/09/19
Time: 12:00
Location: NE Meeting Space, QMB
Participants: Brian Pluss, John Parsons, Connor Haining, James Read, Sam Glendenning, Saif Fardan, Youssef Aloulou, Victor Yu, Adam Munro
Absent: Craig Ramsay
Minutes by Sam Glendenning

**TOPIC**
**Frontend Demo**
JP started off by demoing our progress with the frontend web portal. He demonstrated the homepage with search bar and filters for cost, state, zip code and procedure, along with basic keyword mapping. CH and JR expressed intrigue and posed some keywords to test it with, with some success but irrelevant procedures also being shown in the search suggestions. SG reiterated this was a work in progress. The search results were hard-coded, which CH and JR were made aware of, and they seemed happy with the tabular results that came up.

**TOPIC**
**Version Control**
CH questioned our usage of version control for managing the project. JP pointed him to our Git repository for the frontend, demonstrating the use of branches for different areas of work. CH and JR seemed satisfied with it.

**TOPIC**
**Map Demo**
JP demonstrated his work on implementing a Google Maps API for mapping data results to points on the map. He also showed how he'd figured out how to get text overlays on the pointers, meaning each point can be labelled with a hospital name and cost.

**TOPIC**
**Testing**
JR asked about our testing policy. SG informed him of unit testing being performed on the API. JP mentioned that testing on the frontend is very much on the back burner and will hopefully be tackled in the final sprint.

**TOPIC**
**Further questions**
SG asked CH and JR about the financial figures in the dataset, clarifying if the team's meaning on them is correct. JR confirmed that the average total payments is what the uninsured individual pays and average total payments minus average medicare payments is what the medicare-insured individual pays.

Type: Management Team Meeting
Date: 25/09/19
Time: 12:15
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Brian Pluss, John Parsons, Sam Glendenning, Adam Munro, Victor Yu, Saif Fardan
Absent: Youssef Aloulou
Minutes by Sam Glendenning

**TOPIC**
**Recent Progress**
JP mentioned the use of local database import before actually updating data. JP also mentioned the use of stored procedures on database level. SF mentioned that progress on the search is going well, explaining to CR and BP how the search previously was unoptimised and giving irrelevant results back but now the search has been improved. SF mentioned he is now researching implementing this technology into the frontend. CR seemed happy with the progress.

**TOPIC**
**How the last client meeting went**
JP mentioned the client meeting went well and the client seemed happy. Clients seemed interested in the design and trying out queries based on searches

**TOPIC**
**Progress on map API**
JP explained the progress in the map API and adding pointers of hospitals dynamically to the map. CR asked if this was a complex issue. VY explained the intricacies in calculating the best route between hospital and user location.

**TOPIC**
**Performance on Search**
CR asked how the performance of the search was. JP informed him of problems everyone was having with the silva database server so queries were slow but that localised testing was positive.

**TOPIC**
**Evaluations**
CR reminded the group of the evaluations necessary for the end product and who could carry out these evaluations, explaining the Ethics Approval section on MyDundee had information on this. SG asked for clarification on what sections to best evaluate the end product, and CR mentioned accessibility, usability, performance, how self-explanatory the end product is, etc. The rest of the group took this on board.

**TOPIC**
**Report**
CR reminded the group of the importance of the report and the grade weighting it carries. SG asked if a test table in the report was necessary and CR explained a separate document as an appendix was suitable. BP explained how appendices can be used for reference in the report.

**TOPIC**
**Where we hope to be**
JP explained we hope to have a proper dynamic search done by the end of the week using live data from the database, as well as the distance calculations ready for the map API to display. This would cover the searching, cost limitation and filtering required for the main functional requirements. JP also clarified we hope to show the most recent data first and foremost but also display optional historic data for the same procedure in different years, with a mind to give a price prediction for the next year.

Type: Management Team Meeting (Team Leader)
Date: 27/09/19
Time: 10:30
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, John Parsons
Absent: None
Minutes by John Parsons

**TOPIC**
**Team Attendance**
CR asked if everyone in the team was regularly attending and contributing to the project. JP expressed the opinion that the team was working well together and outlined the roles and responsibilities of each member.

**TOPIC**
**State of the Project**
CR questioned the state of the project and what the team's priorities were. JP informed CR that the product was on-track to meet the base requirements by the next client meeting. CR then asked if there were any additions planned and if there had been considerations towards the report. JP informed him that the report had been started and that SF's python script was an extension that the team aimed to implement.

**TOPIC**
**Ethical Approval**
JP mentioned that the team planned to carry out user testing early in week 3. He questioned CR on the format of the tests and where to submit the ethics declaration form. CR explained that the format could be flexible so long as there was evidence of why an approach was chosen. He then informed JP that a copy of the signed ethics declaration was to be submitted either in hard copy or digitally before testing.

-----------------------------------------------------------------------------------------------------------------

Type: Client Team Meeting
Date: 30/09/19
Time: 10:30
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Connor Haining, James Read, Victor Yu, Sam Glendenning, John Parsons, Adam Munro, Saif Fardan
Absent: Youssef Aloulou
Minutes by Sam Glendenning

**TOPIC**
**How the last sprint went**
JP talked about how the sprint went well, improvements on autocomplete, meeting our targets, developing the search. SF talked about improvements since the last meeting when we searched for the word "ear" and results for "ear" and "heart" came up and now that this is fixed. JR asked how this was fixed and SF talked about how results were prioritised from the dictionary of results based on the specific letters the user had entered, such as exact matches first and then partial matches added later. JP mentioned this was converted from Python to JS.

**TOPIC**
**New Search Demo**
JP demonstrated the autocomplete of keyword to procedure name. The query optimisations went down from 30 seconds to 7 and then views got this down to 1 second for a search. A demo for "Throat, ear and mouth procedures" proved this and JP displayed the map of the closest 10 results, as well as the 10 results in the table. He also demonstrated sorting affecting the updating of the pointers in their prioritisation. JP also showed hardcoded data in the "Further details" page for a given procedure, with map and graphs, along and

plans for developing this to predict future prices. CH said this looked "absolutely brilliant" and would be satisfied with it "being public". An improvement was suggested whereby the column of procedure name, in which the name was all the same, could be removed and used as a header instead. The team agreed, as well as JR. JR also asked if a unit of distance could be displayed. JR also asked how the search box above the results could be used, and JP responded how it is used for string queries only. JR tested this with a number entry and JP responded saying distance is prioritised and then prices. JR believed the single search box to be ambiguous and asked us to elaborate and use user testing to get ideas for improving this. CH backed this up.

**TOPIC**
**User Testing**
JR asked if any user testing had been done. JP responded none yet. JR pressed us to make a move with this, and make use of mostly non-Computing students for test subjects.

**TOPIC**
**Future Plans**
JP talked about documentation to be done as well as finishing the further details page. He also mentioned the basic login ability for an admin but that this was less of a priority compared to the documentation.

**TOPIC**
**Root Website**
JR asked what would happen if the root of the website was accessed, and JP tested that it would show "Forbidden". When asked why, the team elaborated that this was because of spying going on between groups accessing each other's websites. JR and CH seemed disappointed this was going on and allowed us a "free pass" on that one.

**TOPIC**
**User Stories**
JR asked if we elaborated on our athletes as user stories. JP reaffirmed the reasons behind this user group and that we weren't limiting our website to that group of users but that we hadn't revisited this section yet. CH agreed with this but asked us to avoid bias, as we are not meeting expectations for other groups.

**TOPIC**
**Accessibility**
JR asked if we'd thought about accessibility. JP responded with contrasting colours for colour blindness, Bootstrap support for website scaling and responsiveness. JR mentioned that issues were happening with the website on iPhone and asked us to take a look at it due to the increasing use of web access on mobile devices.

Overall, JR and CH seemed very impressed and looked forward to seeing it in its final form on Friday.

--------------------------------------------------------------------------------------------------------------

Type: Management Team Meeting
Date: 30/09/19
Time: 14:30
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Brian Pluss, Victor Yu, John Parsons, Youssef Aloulou, Adam Munro, Saif Fardan
Absent: Sam Glendenning
Minutes by John Parsons

**TOPIC**
**Recent Progress**
JP mentioned the progress since the last meeting. He highlighted the fact that dynamic search had been implemented along with dynamic map markers.

**TOPIC**
**Results of the Client Meeting**
JP informed CR & BP of the client's feedback on the product. He explained that they were very pleased with the demo had no major qualms regarding its functionality.

**TOPIC**
**Evaluations**
CR asked when we were planning to carry out user testing. JP outlined the plan to do testing over the approaching Tuesday and Wednesday.

**TOPIC**
**Future Plans**
JP talked about the aim to tie up the development of new features by the end of Wednesday. He expressed that the reason behind this was due to the importance of the report and presentation.

**TOPIC**
**Question Regarding Appendices**
JP asked how the appendices should be attached to the report. CR responded by saying that they should be pasted at the end of the report in their original format where appropriate (ie paste documents but link gitHub content etc).

---------------------------------------------------------------------------------------------------------------

Type: Management Team Meeting (Team Leader)
Date: 02/10/19
Time: 10:00
Location: NE Meeting Space, QMB
Participants: Craig Ramsay, Brian Pluss, John Parsons
Absent: None
Minutes by John Parsons

**TOPIC**
**Project Progress**
CR asked JP about where he thought the team was in terms of project completion. JP explained that the development was winding down and minor adjustments would be all that coding left to do. JP also remarked that the report and presentation were coming along well.

**TOPIC**
**Questions**
JP asked about the importance of a live demo for the presentation. CR replied by saying that a live demo would be more impressive to the client, but a video would be a good backup. BP also suggested that a video may be good when marking the product post-presentation.

# Appendix B

**Title**: Design ideas and draft work

**Relevance**: Gives informal information on the ideas the team had when designing the project, as well as draft pieces of work and brainstorms when figuring out problems and issues

## Requirements

### Functional

- The user shall be able to input search terms as part of their queries in order to retrieve results of procedures that cover those search terms. At a minimum, relevant results shall be returned for a search by injury or body area
- The user shall be able to order results of queries by price, both in ascending and descending order
- The user shall be able to sort and filter results of queries by location, either by inputting their location or allowing their location to be retrieved from their IP address
- The system shall return results for search queries, each one containing a practice name, location, distance from user, price and offered procedure(s)
- Search results shall be given restrictions and limits on distance from user and price range
- The user shall consult a map of locations, with placeholders for price and practice name
- The system shall resolve specific queries regarding a particular illness or injury to an umbrella category of procedures that practices can be queried on
- The user shall specify conditions through which to filter data before and after searches are executed, in order to limit the amount of data displayed and ensure more relevant data is displayed as much as possible

### Non-functional

- The system should give hospitals and practices a rank in the user's search results in descending order. This rank may be based on, but not limited to, number of discharges by practice, cost of procedure by practice, distance between patient and practice, etc.
- The system may learn from the user's past searches and queries based on cookie information in their computer and use this data to provide more relevant results on future searches
- The site may be more accessible to the end user, such as the provision of colour-blind and contrast options, scalability of the site (zooming, window resizing, etc.)
- The system may limit the amount of data that is returned to the user to prevent slowdown or unnecessary bandwidth usage

## User Stories

- As a patient, I can search by my injury so that I can find practices that offer care catering to me and my injury
- As a patient, I can filter results of practices by location so that I can find care within a reasonable distance of me
- As a patient, I can order results of practices by price so that I can find the most affordable care for me

- As a patient, I can view a map of practices and prices to geographically consider the best choice of care for me
- As a patient, I can restrict results based on limits on price and distance to practice so that I can find care based on my financial and travel budgets
- As a patient, I can view care options available to me such that I can compare them specifically on price and location in order to find the best option available to me
- As a patient, I can browse practices local to me so as to gain an understanding of procedures available in my location
- As a patient, I can search by the exact procedure I want in order to know all the places in the United States where the procedure is offered and the respective prices of care
- As a system admin, I can add and remove information on practices and procedures offered so that the most up to date information is available to patients

-----------------------------------------------------------------------------------------------------------------------------

**Use Cases**

**USE CASE 1: Search by procedure**

**Actors**

The following actors are involved in this use case:

- Patient

**Brief Description**

Allow the patient to search the web frontend for a specific procedure to retrieve results of practices that offer that procedure. This queries the database for a DRG definition that covers that procedure. Results are displayed in a table, filtered based on the patient's requirements

**Basic Flow of Events**

The use case begins when the patient inputs a procedure name into the search box and initiates a search

| | |
|---|---|
| Patient : | Inputs search parameter |
| System: | Search for that string in the procedure tags of each DRG definition |
| System: | For each one, add it to the list of returned results |
| System: | Displays results in table based on filters selected by patient |

## USE CASE 2: Filtering results

**Actors**

The following actors are involved in this use case:

- Patient

**Brief Description**

Allow the patient to filter received results of practices and procedures based on location and/or price

**Basic Flow of Events**

The use case begins when the patient uses the sliders next to the results table to place limits on what results are displayed

**By distance:**

| | |
|---|---|
| Patient : | Limiting distance to practice to within 500km |
| System: | Web results table removes results of practices greater than 500km from patient |
| System: | Displays above results in table |

**By price:**

| | |
|---|---|
| Patient : | Limiting cost of procedure to within $10,000 |
| System: | Web results table removes results of procedures costing more than $10,000 |
| System: | Displays above results in table |

----------------------------------------------------------------

## USE CASE 3: Search by injury/illness

**Actors**

The following actors are involved in this use case:

- Patient

**Brief Description**

Allow the patient to search the web frontend for a specific injury or illness to retrieve results of practices that offer procedures to address that injury or illness. This queries the database for a DRG definition that covers that procedure by using the tags of that procedure. Results are displayed in a table, filtered based on the patient's requirements

**Basic Flow of Events**

The use case begins when the patient inputs an injury or illness into the search box and initiates a search

Patient : Inputs search parameter
System: Search for that string in the injury/illness tags of each DRG definition
System: For each one, add it to the list of returned results
System: Displays results in table based on filters selected by patient

--------------------------------------------------------------

## USE CASE 4: Search by location

**Actors**

The following actors are involved in this use case:

- Patient

**Brief Description**

Allow the patient to search the web frontend for a specific location, through which the nearest practices and the procedures they offer will be displayed to the user. This would allow for the patient to see local or nearby practices and gain an understanding of the healthcare nearby to them.

**Basic Flow of Events**

The use case begins when the patient inputs a location into the search box

Patient : Inputs location
System: Search for that string in the city locations of the results
System: For each one, add it to the list of returned results
System: Displays results in table in ascending order of distance from patient

### USE CASE 5: Adding new information to records

**Actors**

The following actors are involved in this use case:

- System admin

**Brief Description**

Allow the system admin to add new records to the dataset for new hospitals or practices or amend existing ones to add or remove procedures offered to patients. This allows for a patient to receive the most up to date information on what healthcare is available to them.

**Basic Flow of Events**

The use case begins when the admin logs in to the web portal.

| | |
|---|---|
| Admin : | Selects Insert/Update Data |
| Admin: | Fills in form for adding figures for a procedure |
| System: | Resolves the hospital and procedure definitions to their respective ID numbers |
| System: | Inserts the data into the relevant table |

-----------------------------------------------------------------------------------------------------------

# User Personas

**Joe**

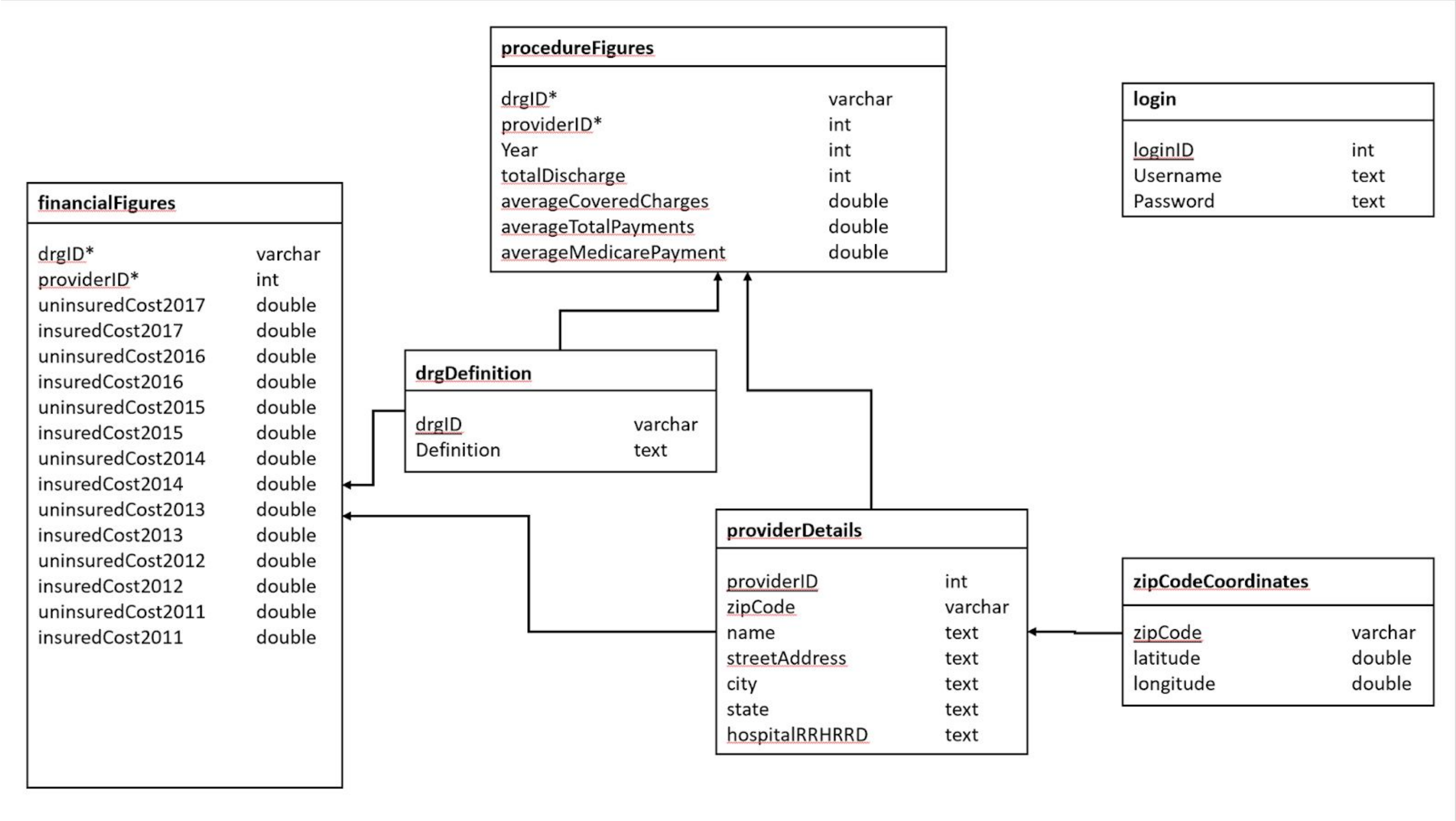
Demographics: Employed; Age: 18-35; Single; Male<br>
Location: Boston, USA<br>
Job: Delivery driver<br>
Family: lives with his cat<br>
Background: Joe works a simple and modest job as he only needs to provide for himself. He has a passion for MMA combat and fights during his spare time.<br>
Needs: Due to his hobby which preoccupies him most of his time, Joe often finds himself needing treatment for his injuries. He requires to find the closest and most cost-efficient hospital able to cover the price of his treatments/procedures. Joe isn't insured due to his minimum wage job.

**Jaylen**

Demographics: Unemployed; Age: 18-23; Single; Male<br>
Location: New York, USA<br>
Job: unemployed<br>
Family: lives with parents and younger brother<br>
Background: Jaylen is a young basketball prospect who's training daily to become an NBA star.<br>
Needs: Jaylen realises that career-ending injuries are part of his journey into making it to the NBA. He looks to find a hospital able to treat any potential injuries that happen to him which his parents cannot cover.

**Sophia**
Demographics: Employed; Age: 25-40; Married; Female
Location: Philadelphia, USA
Job: BMX athlete
Family: lives with her newlywed
Background: Sohpia is one of the top BMX riders in the country who participates in many contests and races.
Needs: Due to the danger of the sport, Sohpia has broken several bones already. She has had to seek private medical treatment for her injuries which has delayed her honeymoon. Sophia requires a hospital that can treat her injuries at an affordable price and close distance.

### Design for Database



| procedureFigures | |
|---|---|
| drgID* | varchar |
| providerID* | int |
| Year | int |
| totalDischarge | int |
| averageCoveredCharges | double |
| averageTotalPayments | double |
| averageMedicarePayment | double |

| login | |
|---|---|
| loginID | int |
| Username | text |
| Password | text |

| financialFigures | |
|---|---|
| drgID* | varchar |
| providerID* | int |
| uninsuredCost2017 | double |
| insuredCost2017 | double |
| uninsuredCost2016 | double |
| insuredCost2016 | double |
| uninsuredCost2015 | double |
| insuredCost2015 | double |
| uninsuredCost2014 | double |
| insuredCost2014 | double |
| uninsuredCost2013 | double |
| insuredCost2013 | double |
| uninsuredCost2012 | double |
| insuredCost2012 | double |
| uninsuredCost2011 | double |
| insuredCost2011 | double |

| drgDefinition | |
|---|---|
| drgID | varchar |
| Definition | text |

| providerDetails | |
|---|---|
| providerID | int |
| zipCode | varchar |
| name | text |
| streetAddress | text |
| city | text |
| state | text |
| hospitalRRHRRD | text |

| zipCodeCoordinates | |
|---|---|
| zipCode | varchar |
| latitude | double |
| longitude | double |

# Appendix C

**Title**: Sprint reviews and retrospectives

**Relevance**: This document holds formal notes of the team's weekly sprint reviews and retrospectives. It gives an idea of the progress the team made with each week, as well as gauging how the team felt about their efforts.

**Thoughts on Sprint 1**

- Daily stand-ups took place every weekday between 9:30am and 10am
- Attendance was generally good. Youssef was absent from the QMB for a couple of days but did some work at home
- At the end of the sprint, out of the 23 tasks in the sprint backlog, 1 was at 'Ready/On Hold' status, 3 were in 'Doing' and 19 were in 'Done'
- This makes our sprint burndown generally quite good, with only a few tasks overestimated

**Review**

What went well - attendance is good overall, stand up attendance pretty good, designation of tasks was good, communication was good

What didn't go well - database import could have been studied better so that it could have been imported quicker, planning overall could have been better regarding backlog, missed Youssef for a couple of days

What could we have done to improve this? - studied data a little better, sanitised it better, dedicate more time to planning, keep it up

---

**Thoughts on Sprint 2**

- Clients seemed very impressed
- Most impressed in the performance enhancements
- Also like the keyword mapping as no other group is doing it
- Database problems continued but we persevered
- Team incorrectly assumed the data was all the same format
- Could have in hindsight put more effort into checking this but there wasn't much we could have foreseen
- Since last week, we've gone from no dynamic search and map with basic JSON data reading to now having all the base requirements and most of the detail view done
- Also implemented login, began work on inserting and updating
- Keyword generator is done
- Also made good progress on report, 5K words and 9 pages

**Review**

What went well - clients were impressed, sprint backlog all done, see above

What didn't go well - database was an issue, issues between JS and Python required Python to be converted, more time than we wanted spent on JS and table problems

What could we have done to improve this? - analysed dataset more thoroughly, more research into Python and JS compatibility

**Thoughts on Sprint 3**

- Client was very happy with the presentation and demo
- Team did what they could to wrap up the project to a good working state
- Bit of a sprint finish in places
- Happy with the documentation we have
- Believe we hold our own against other groups
- Database finally got to an acceptable state with fast query times
- In order to finish up, had to implement a couple of hacky solutions for things, e.g. cookies to pass a value between javascript and php instead of other more robust methods.
- Overall, team did well to estimate what we could and couldn’t achieve by the end of the sprint

**Review**

What went well - clients were happy, project wrapped up well, database sorted
What didn’t go well - a couple of programming solutions not the best in the long-term but work for now, asynchronous search functionality isn’t to the best code practice
What could we have done to improve this? - researched our approach to asynchronous functionality better at the start of the project, planned the reading and writing of files beforehand

**Review of all Sprints**

Team is happy with what they achieved. We presented a working demo and strong presentation to the class and client and are pleased with the speed and functionality of our search, as well as the implementation of data insertion by an admin user. However, project structure and version control was not the best on the frontend and we often had multiple team members working on the same files at the same time, which caused conflict with Git. We worked hard to wrap this up so that master is in an acceptable state. The API’s repo was better organised and does not suffer from conflicts.
We did well to keep on top of the documentation and held regular stand-ups and reviews and retrospectives. Overall, the progress made in 3 weeks was good, although we could have done with more time to make the project exactly what we wanted it to be.

# Appendix D

**Title**: Summaries of daily progress

**Relevance**: This document holds informal summaries by John Parsons about daily work the team underwent.

**Daily Notes by John Parsons**

**Week 1**

Aim: get dev environment set up and have a prototype application in the works

**16/9/19**

Decide on the following:

Development environment?

- Web-based (PHP, HTML, JS, Bootstrap)
- Framework?

Methodology?

- Agile **9:30** standups

Requirements Doc

**17/9/19**

Setup dev environment

Get working on website

Import data into DB

**18/9/19**

Continue working on website

Google Map API

Continue importing data into DB

**19/9/19**

Implement dummy search with loading screen

Continue importing data into DB (resolve problem with importing the second half of the data)

Work on the python similar word list

**20/9/19**

Integrate results table with dummy search

Import data locally and create dump file

Continue work on the python similar word list

**Week 2**

Aim: push on with application development

**23/9/19**

Split up table data

Populate table with query data

**24/9/19**

Work on frontend structure

Mapped words to procedures

JS animations and loading page

**25/9/19**

Implemented proper loading screen

Results table populates on API request

Worked on detail page

Worked on stored procedures

**26/9/19**

Frontend - charts on detail page, about page and contact page

Dynamic search querying

Passing JS variables to python script

**27/9/19**

Speed up query times

Update API

Create admin page

**Week 3**

Aim: refine searching, report, presentation

**30/9/19**

Distance now calculated from user location

Results map updates the top 10 pointers dynamically

Autocomplete function converted to JS and now more accurate

**01/10/19**
Progress on the admin dashboard
Started on the presentation
Created plan for user testing

**02/10/19**
Carried out user testing
General frontend improvement

**03/10/19**
Prepared for presentation
Finished off development
Got database to good state

**04/10/19**
Practised for presentation
Prepared demo
Smashed it

**05/10/19**
Finalised documentation
Submitted report

# Appendix E

**Title**: User Manual for *CompareTheHealthcare.com*

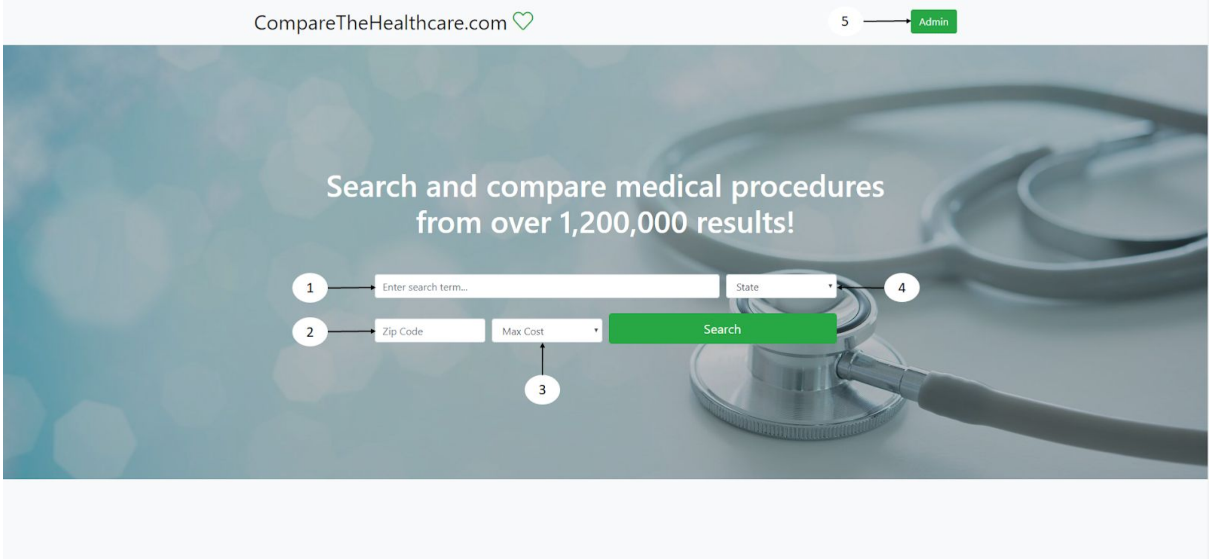
**Relevance**: This document contains a formal user guide and descriptions on how to use the web portal and what a user can get out of it. This includes screenshots of the portal in various use cases, details on the information provided and the purposes behind the software.

## Welcome

Thank you for choosing *CompareTheHealthcare.com*. This manual should act as a guide for using the web portal and searching for the procedure that's right for you. *CompareTheHealthcare.com* gives financial and locational information on over 1.2 million individual procedural results, helping you to shop around for the deal that's right for you.

## Searching

When launching the website, you will get this page:

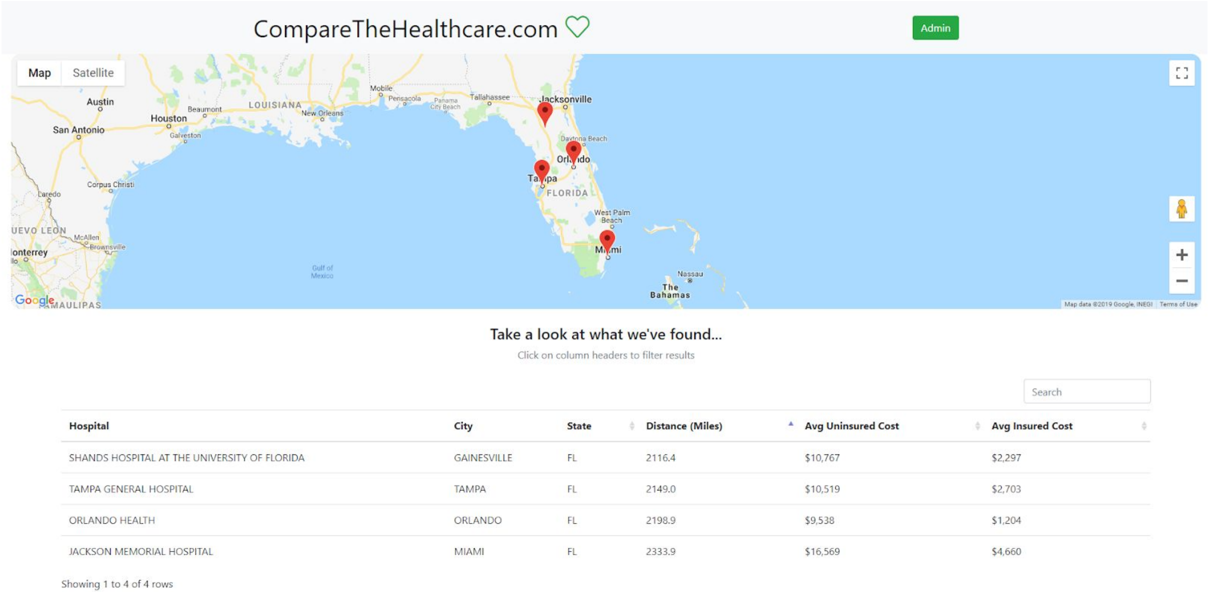


1. Enter procedure or body part term
2. Enter your zip code
3. Enter maximum cost willing to pay for procedure
4. Enter state where you would like to undertake the procedure
5. Admin login page

The search can be used for browsing procedures by name or keyword, allowing for specification on which state to search for. When entering a search term, allow for your keyword to resolve to a result from the dropdown. If nothing is suggested, your search will not display any results. Entering a search state and your zip code and cost you are willing to pay are entirely optional but may help you narrow down your results.

Allowing the website to access your location is entirely optional but highly recommended. Your browser will ask to use your location when you first visit the site: press "Allow" or "Yes" to let this go ahead. The purpose of this is to find the closest procedures for you and is used in the ranking of results.

## Results



CompareTheHealthcare.com

Take a look at what we've found...

Click on column headers to filter results

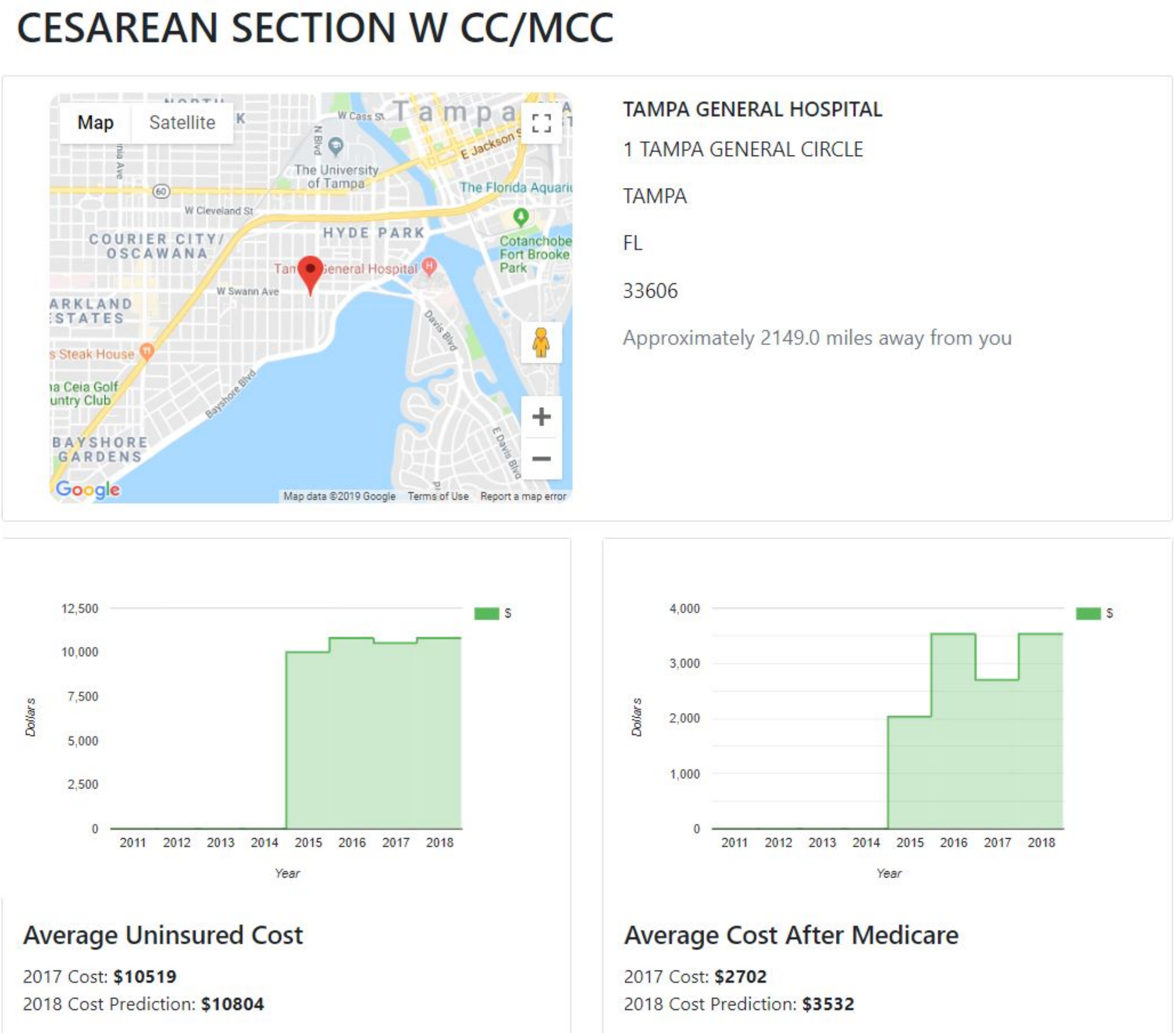
Search

| Hospital | City | State | Distance (Miles) | Avg Uninsured Cost | Avg Insured Cost |
|---|---|---|---|---|---|
| SHANDS HOSPITAL AT THE UNIVERSITY OF FLORIDA | GAINESVILLE | FL | 2116.4 | $10,767 | $2,297 |
| TAMPA GENERAL HOSPITAL | TAMPA | FL | 2149.0 | $10,519 | $2,703 |
| ORLANDO HEALTH | ORLANDO | FL | 2198.9 | $9,538 | $1,204 |
| JACKSON MEMORIAL HOSPITAL | MIAMI | FL | 2333.9 | $16,569 | $4,660 |

Showing 1 to 4 of 4 rows

When a search is carried out, a successful one returns results in a format as above for a search for “Caesarean Section”. A table displays a list of results in ascending order of distance from the user’s location, along with details for the hospitals providing that procedure and the average uninsured and insured price of undergoing that procedure in the last year. The map above the results shows the geographical location of the hospitals currently being displayed in the list, relative to where the user is located.

## Further Details



CESAREAN SECTION W CC/MCC

TAMPA GENERAL HOSPITAL
1 TAMPA GENERAL CIRCLE
TAMPA
FL
33606
Approximately 2149.0 miles away from you

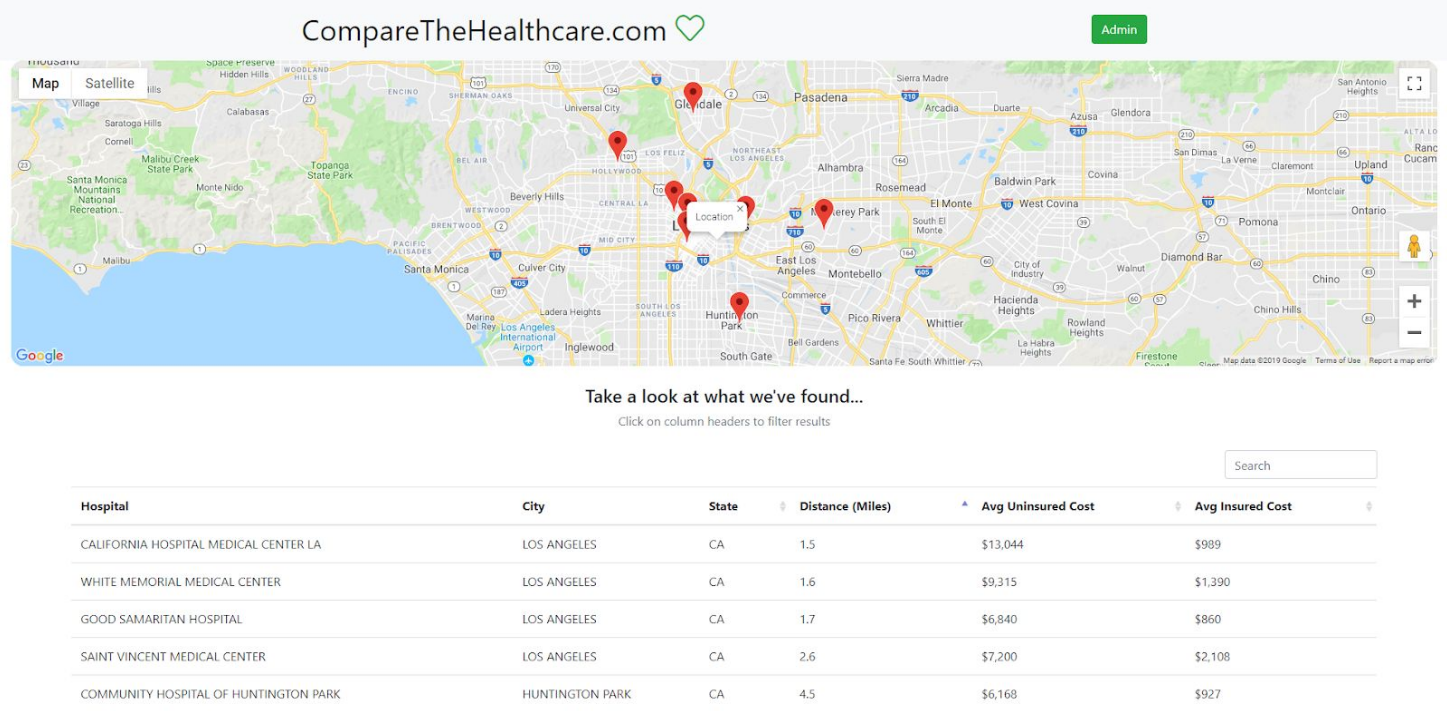
Average Uninsured Cost
2017 Cost: **$10519**
2018 Cost Prediction: **$10804**

Average Cost After Medicare
2017 Cost: **$2702**
2018 Cost Prediction: **$3532**

Clicking on one of the results in the table brings up further details of the procedure at that hospital. The address of the hospital is displayed alongside a localised map of that hospital. Graphs below give a visual timeline of how the cost of that procedure has changed over the years, allowing for trends to be drawn from the data. Last year's cost of the procedure is displayed along with a prediction for the cost in the next year.

### Example Usage

You are a Californian citizen insured by Medicare suffering from some pain in your chest. By entering “chest” into the search, a list of search suggestions are displayed. You click on one of them, which mentions procedure “313 - CHEST PAIN” - exactly what you need! You have allowed your location so you need only enter “ANY” state and you specify your maximum budget as $3000. Upon hitting search, you receive the following page.



Take a look at what we've found...

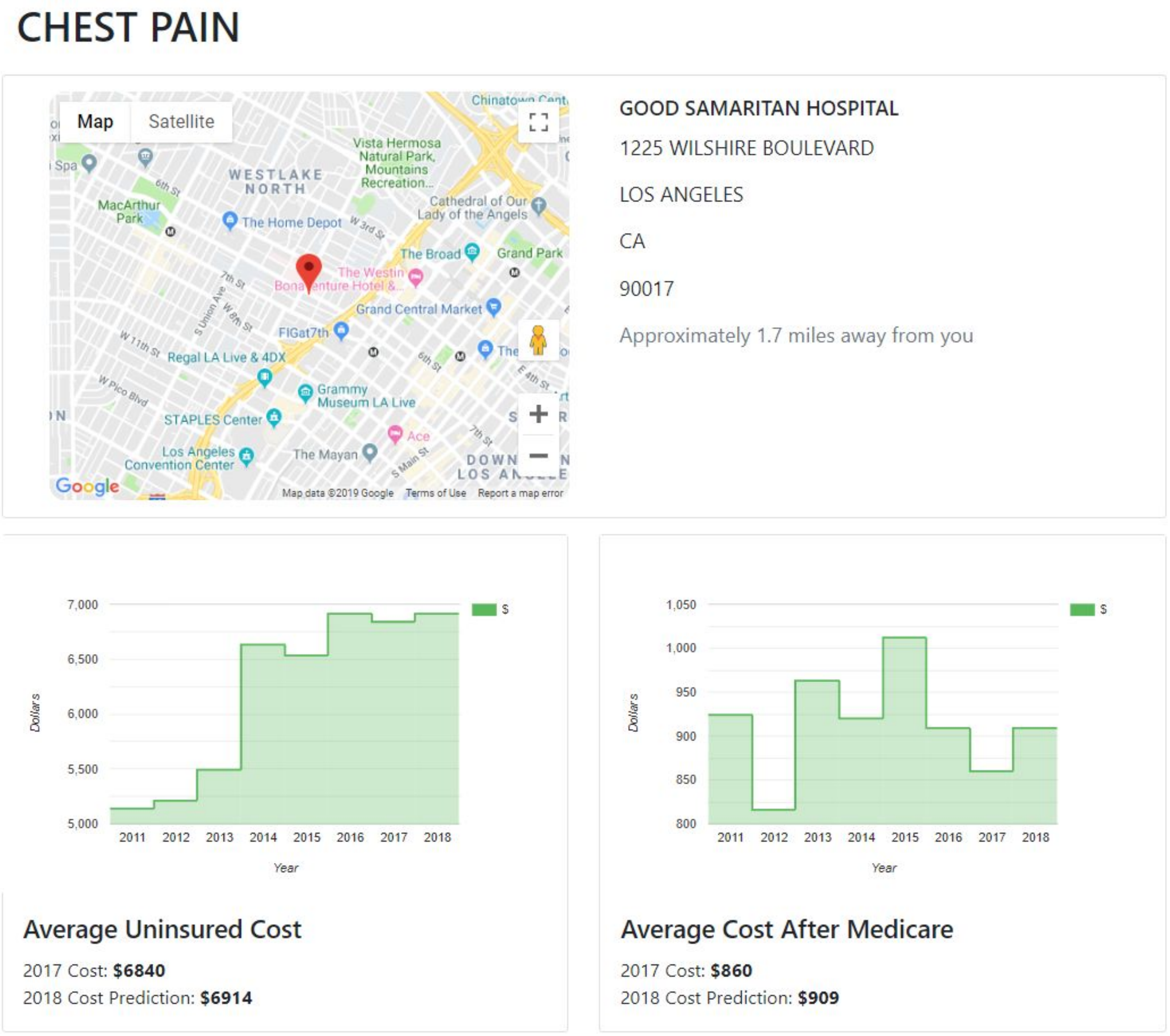
Click on column headers to filter results

Search

| Hospital | City | State | Distance (Miles) | Avg Uninsured Cost | Avg Insured Cost |
|---|---|---|---|---|---|
| CALIFORNIA HOSPITAL MEDICAL CENTER LA | LOS ANGELES | CA | 1.5 | $13,044 | $989 |
| WHITE MEMORIAL MEDICAL CENTER | LOS ANGELES | CA | 1.6 | $9,315 | $1,390 |
| GOOD SAMARITAN HOSPITAL | LOS ANGELES | CA | 1.7 | $6,840 | $860 |
| SAINT VINCENT MEDICAL CENTER | LOS ANGELES | CA | 2.6 | $7,200 | $2,108 |
| COMMUNITY HOSPITAL OF HUNTINGTON PARK | HUNTINGTON PARK | CA | 4.5 | $6,168 | $927 |

You are presented with a list of hospitals offering the procedure you need, in ascending order of distance from your location. Comparing the results and sorting the table by clicking on the column headers, you can see that the Good Samaritan hospital, 1.7 miles away from you is perfect and affordable at $860.

You click on that result and are presented with the following page:



## CHEST PAIN

GOOD SAMARITAN HOSPITAL
1225 WILSHIRE BOULEVARD
LOS ANGELES
CA
90017
Approximately 1.7 miles away from you

### Average Uninsured Cost

2017 Cost: **$6840**
2018 Cost Prediction: **$6914**

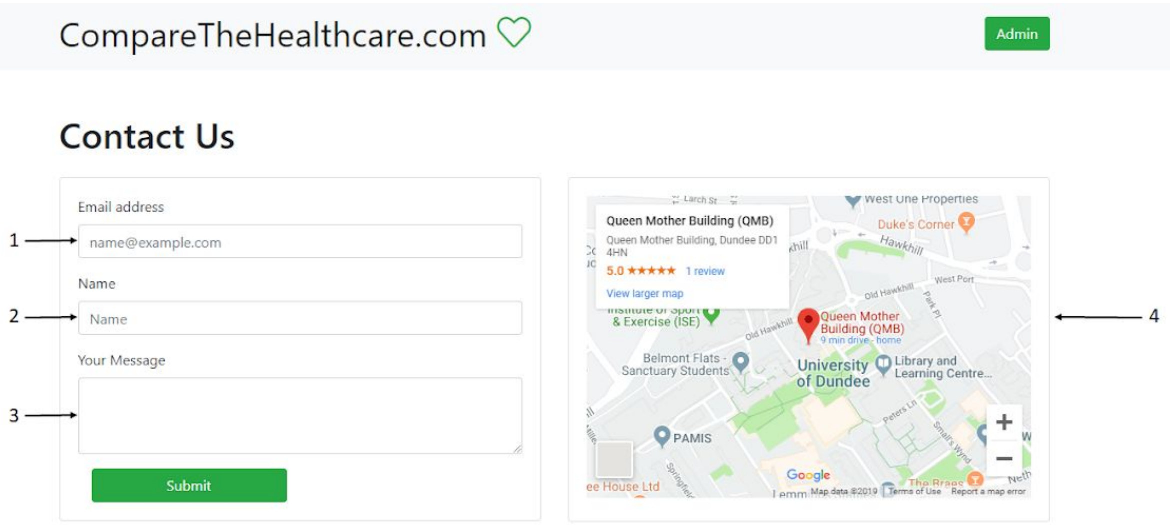
### Average Cost After Medicare

2017 Cost: **$860**
2018 Cost Prediction: **$909**

Here you can see the exact address and location of the hospital. Below are graphs of the change in price of this procedure at this hospital over time, allowing you to visualise trends in the data. The graphs also show a prediction for what this price will be in the next year.

## Contact Page

Clicking the “Contact” button at the bottom of each page will send the user to this page:



CompareTheHealthcare.com

Contact Us

Email address

name@example.com

Name

Name

Your Message

Admin
Submit
1
2
3
4
Queen Mother Building (QMB)
Queen Mother Building, Dundee DD1 4HN
5.0 ★★★★★ 1 review
View larger map
West One Properties
Duke's Corner
Hawkhill
Institute of Sport & Exercise (ISE)
Queen Mother Building (QMB)
9 min drive · home
Belmont Flats - Sanctuary Students
University of Dundee
Library and Learning Centre...
PAMIS
Google
Map data ©2019 Terms of Use Report a map error

1. Enter user’s email address
2. Enter user’s name
3. Enter user’s message
4. Map showing the location of “CompareTheHealthcare.com”

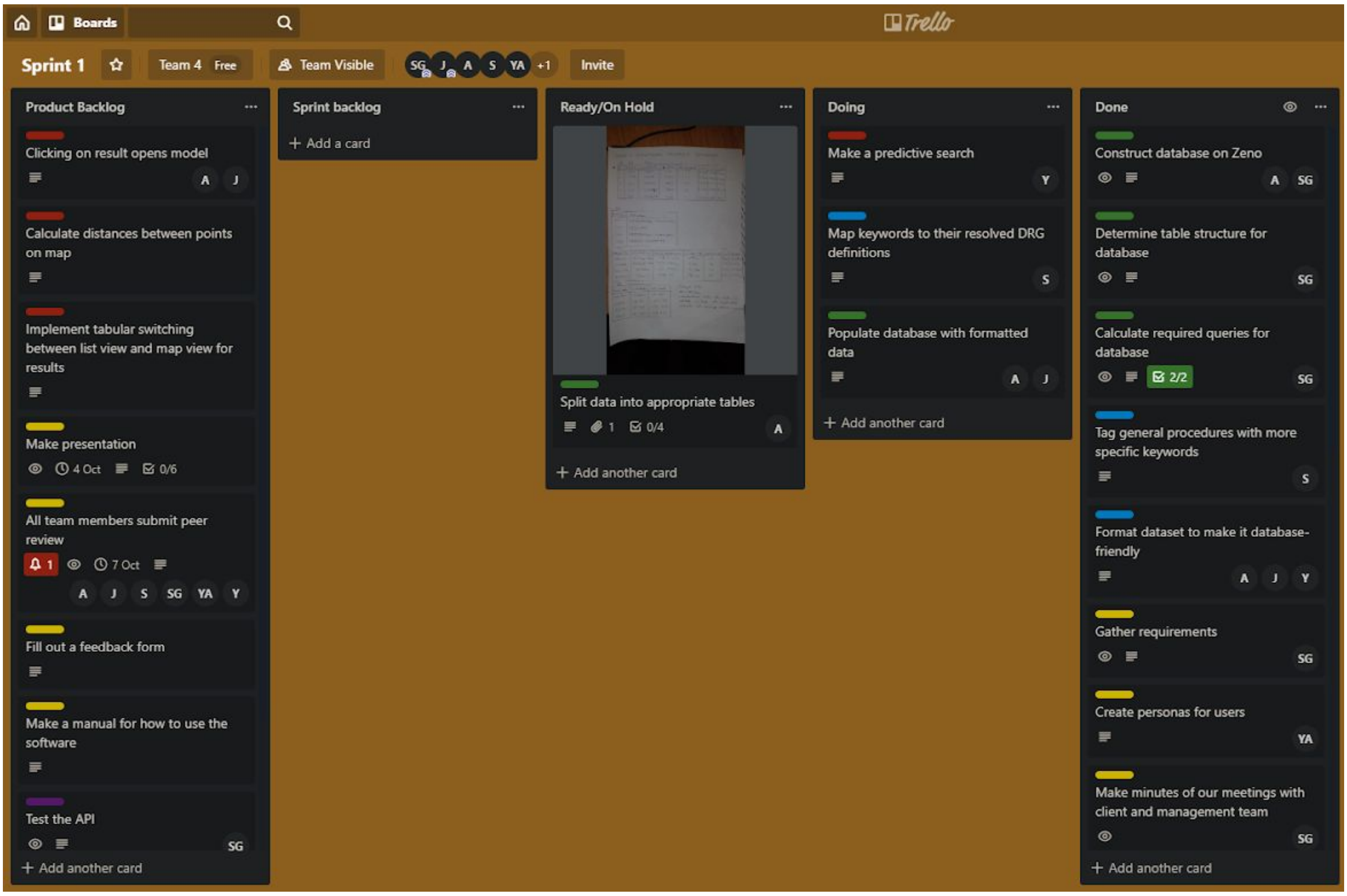
The contact page can be used for contacting the website administrators. It is useful for asking website questions, reporting issues and suggesting features you’d like to see in the website.

# Appendix F

**Title**: Sprint Boards & Backlogs

**Relevance**: The team relied on the Trello online portal for managing and designing tasks and cards to do and assigning team members to them. A board for each sprint was used to prevent cluttering and lists were used for product backlogs, sprint backlogs and sections for tasks on hold, ready to be started, currently in progress and done.

## Sprint 1



**Product Backlog**
- Clicking on result opens model (A, J)
- Calculate distances between points on map
- Implement tabular switching between list view and map view for results
- Make presentation (4 Oct, 0/6)
- All team members submit peer review (1, 7 Oct; A, J, S, SG, YA, Y)
- Fill out a feedback form
- Make a manual for how to use the software
- Test the API (SG)
- + Add another card

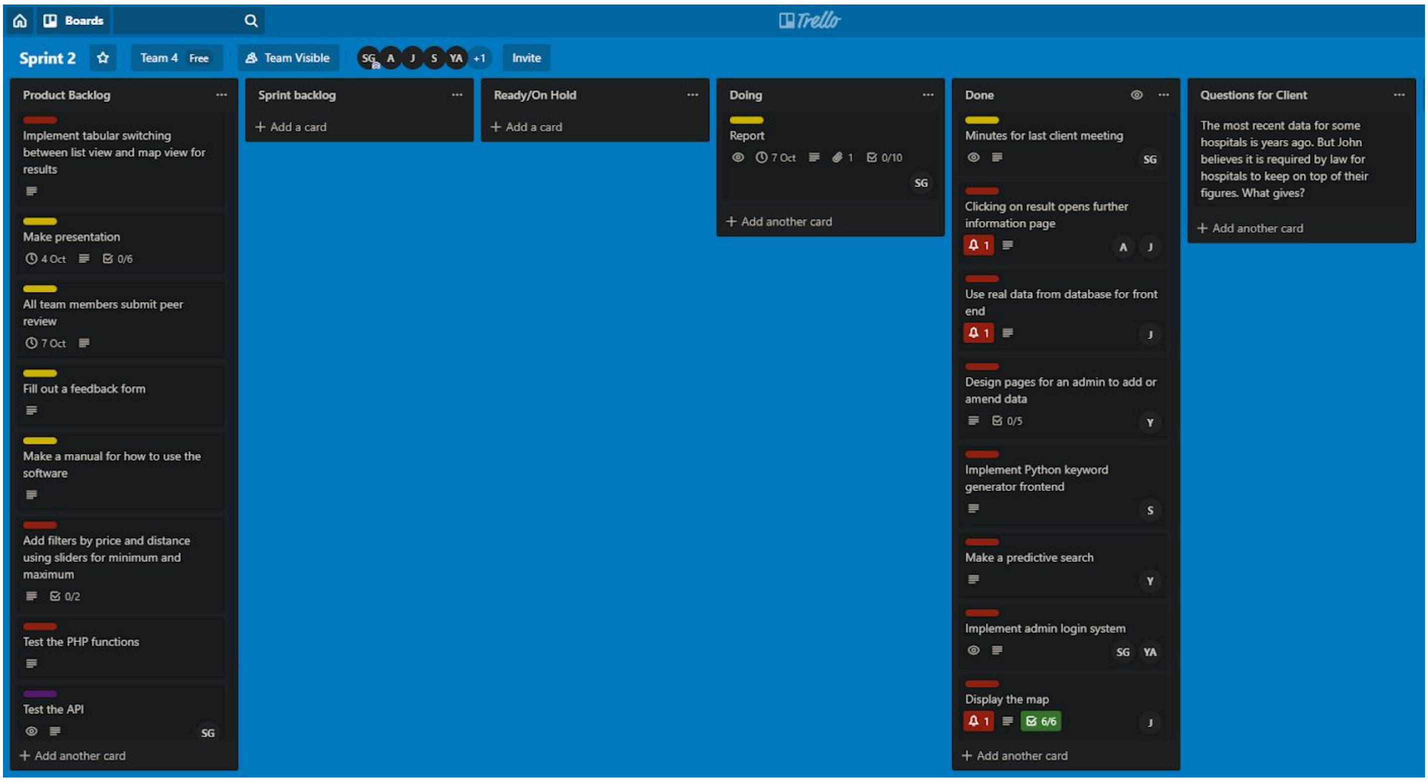
**Sprint backlog**
- + Add a card

**Ready/On Hold**
- Split data into appropriate tables (1, 0/4; A)
- + Add another card

**Doing**
- Make a predictive search (Y)
- Map keywords to their resolved DRG definitions (S)
- Populate database with formatted data (A, J)
- + Add another card

**Done**
- Construct database on Zeno (A, SG)
- Determine table structure for database (SG)
- Calculate required queries for database (2/2; SG)
- Tag general procedures with more specific keywords (S)
- Format dataset to make it database-friendly (A, J, Y)
- Gather requirements (SG)
- Create personas for users (YA)
- Make minutes of our meetings with client and management team (SG)
- + Add another card

## Sprint 2



Product Backlog

- Implement tabular switching between list view and map view for results
- Make presentation — 4 Oct — 0/6
- All team members submit peer review — 7 Oct
- Fill out a feedback form
- Make a manual for how to use the software
- Add filters by price and distance using sliders for minimum and maximum — 0/2
- Test the PHP functions
- Test the API — SG

Sprint backlog

Ready/On Hold

Doing

- Report — 7 Oct — 1 — 0/10 — SG

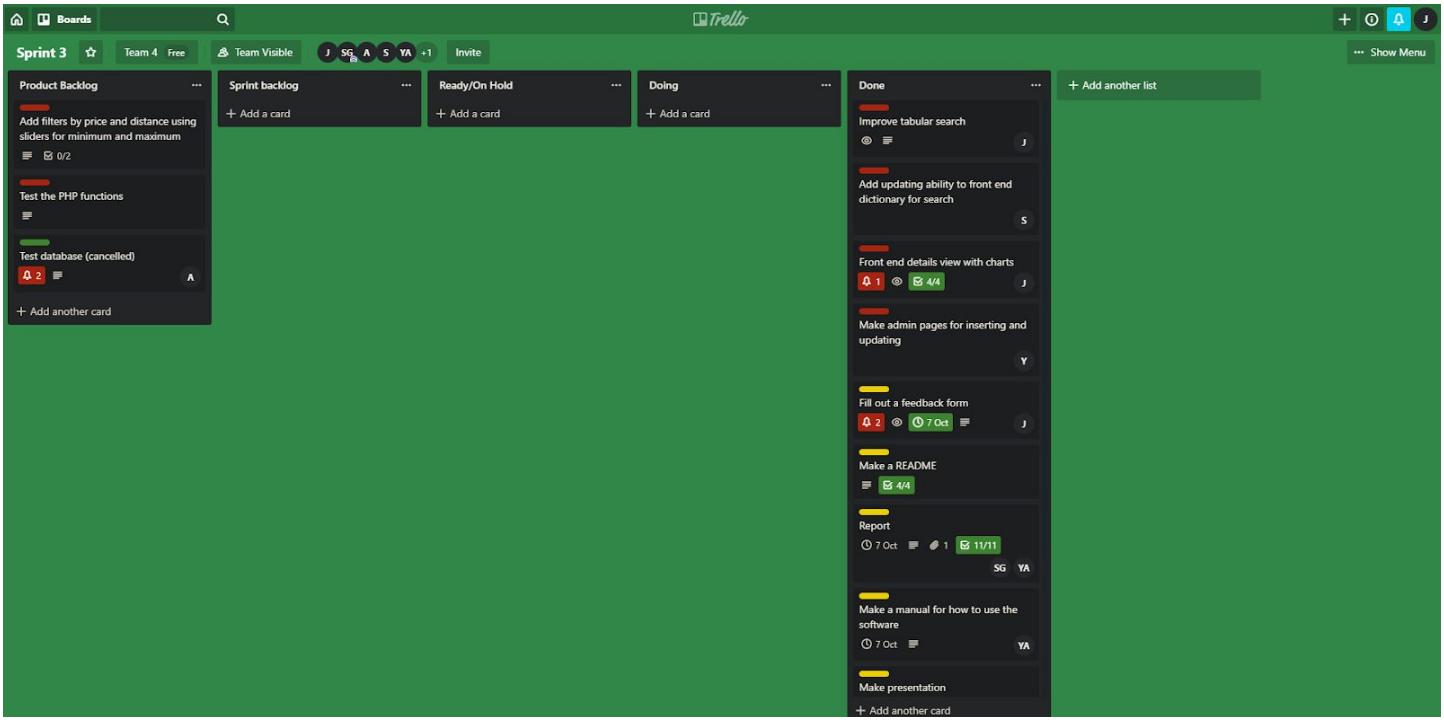
Done

- Minutes for last client meeting — SG
- Clicking on result opens further information page — 1 — A J
- Use real data from database for front end — 1 — J
- Design pages for an admin to add or amend data — 0/5 — Y
- Implement Python keyword generator frontend — S
- Make a predictive search — Y
- Implement admin login system — SG YA
- Display the map — 1 — 6/6 — J

Questions for Client

- The most recent data for some hospitals is years ago. But John believes it is required by law for hospitals to keep on top of their figures. What gives?

## Sprint 3



Product Backlog

- Add filters by price and distance using sliders for minimum and maximum — 0/2
- Test the PHP functions
- Test database (cancelled) — 2 — A

Sprint backlog

Ready/On Hold

Doing

Done

- Improve tabular search — J
- Add updating ability to front end dictionary for search — S
- Front end details view with charts — 1 — 4/4 — J
- Make admin pages for inserting and updating — Y
- Fill out a feedback form — 2 — 7 Oct — J
- Make a README — 4/4
- Report — 7 Oct — 1 — 11/11 — SG YA
- Make a manual for how to use the software — 7 Oct — YA
- Make presentation

# Appendix G

**Title**: User Evaluation Answers to Questionnaire

**Relevance**: External evaluations of the system's search functionality were carried out by multiple individuals. These anonymised evaluations are useful in giving an idea of how the system would be received upon a full release, as well as gaining insight and opinions of the system from those not involved in the team or experienced in computing.

**Preface**

The questionnaire provided to the users can be found in Appendix J.
Statistics of results of questionnaire can be found at the bottom of the appendix.

**Name:** User B
**Date:** 1st October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - Sort/filter of costs
4. **How likely would you use the website if it were relevant to you?** - 4: Very likely

**Scenario 1**

1. **Write your exact search term below** - CHEST PAIN - TEXAS
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 313 - CHEST PAIN
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - The hospitals of Providence Transmountain Campus, $3431

**Scenario 2**

1. **Write your exact search term below** - HEAD
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - 129 - MAJOR HEAD & NECK PROCEDURES W CC/MCC OR MAJOR DEVICE
3. **Did you manage to find a procedure to help you?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - Pennsylvania Hosp of the Univ of PA HEALTH SYS, $1934

**General Questions 2**

1. **How easy was it to navigate through the website?** - 4: Easy
2. **Which features did you like about our website?** - Speed, presentation/visual, maps, contact details of hospital
3. **What would you most like to see improved about this website?** - Heading - "Results", Filter/sort function indicated
4. **Anything you disliked about the website?** - No
5. **Any further comments?** - No

**Name:** User D
**Date:** 1st October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 5: Extremely appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - Short loading time, sleek looking website, easy to use
4. **How likely would you use the website if it were relevant to you?** - 5: Extremely likely

**Scenario 1**

1. **Write your exact search term below** - CHEST PAIN
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 313 - CHEST PAIN
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - The hospitals of Providence Transmountain Campus

**Scenario 2**

1. **Write your exact search term below** - CONCUSSION
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - 088 CONCUSSION W MCC
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price** - No results

**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Simplicity, straight to the point and accurate prediction text
3. **What would you most like to see improved about this website?** - Fix the “no results” that I received, fix the admin login issue (researcher note: a bug occurred during this evaluation where the admin login allowed any credentials)
4. **Anything you disliked about the website?** - Nope
5. **Any further comments?** - N/A

----------------------------------------------------------------------------------------------------------------------

**Name:** User C
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 1: Too little information
3. **What further information could be provided?** - Information of procedures should be listed. Private hospitals could be provided
4. **How likely would you use the website if it were relevant to you?** - Very likely

**Scenario 1**

1. **Write your exact search term below** - Myocardial Infarction
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - 282 - ACUTE MYOCARDIAL INFARCTION, DISCHARGED ALIVE W/O CC/MCC
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - TEXAS HEALTH PRESBYTERIAN HOSPITAL DALLAS - $5409

**Scenario 2**

1. **Write your exact search term below** - Subarachnoid Haemorrhage
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - Nothing
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price** -

**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Very easy to search. Find pricing for both insured and uninsured helpful. Can localise to state and by distance
3. **What would you most like to see improved about this website?** - Could have listed pricing high to low. Aesthetic procedures. Private hospitals. Reviewing of surgeons (see BAAPS)
4. **Anything you disliked about the website?** - Too simplistic for a health professional. More info on procedures
5. **Any further comments?** - Further procedures needed (plastics, dermatology, etc.)

------------------------------------------------------------------------------------------------------------------------

**Name:** User H
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - None
4. **How likely would you use the website if it were relevant to you?** - 4: Very likely

**Scenario 1**

1. **Write your exact search term below** - Vagina, cervix & vulva procedures
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - the correct one
3. **Did your search run as expected?** - No
4. **If you said Yes to the above, write hospital you selected and for which price** -

**Scenario 2**

1. **Write your exact search term below** - Faint, head
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - Major head & neck procedures
3. **Did you manage to find a procedure to help you?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - Baylor University, $4762

**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Very nice frontend
3. **What would you most like to see improved about this website?** - Loading times
4. **Anything you disliked about the website?** - No
5. **Any further comments?** - No

--------------------------------------------------------------------------------------------------------------------

**Name:** User A
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - None
4. **How likely would you use the website if it were relevant to you?** - 3: Somewhat likely

**Scenario 1**

1. **Write your exact search term below** - chest
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 313 - CHEST PAIN
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - LAKE POINTE MEDICAL CENTER, $4648

**Scenario 2**

1. **Write your exact search term below** - conc
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - 088 - CONCUSSION W MCC
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price** -

**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Loading wheel
3. **What would you most like to see improved about this website?** - Search bar should move away when results come up
4. **Anything you disliked about the website?** - No
5. **Any further comments?** - No

--------------------------------------------------------------------------------------------------------------------

**Name:** User F
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - Phone numbers
4. **How likely would you use the website if it were relevant to you?** - 5: Extremely likely

**Scenario 1**

1. **Write your exact search term below** - Chest pain - any state
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 163 - MAJOR CHEST PROCEDURES W MCC
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - Physicians Regional Medical Center

**Scenario 2**

1. **Write your exact search term below** - Head hurt, then concussion when that didn't work
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search -** Concussion followed by weird abbreviations
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price -**

**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Clean UI, fast search
3. **What would you most like to see improved about this website?** - Some search terms, remove confusing elements
4. **Anything you disliked about the website?** - As above
5. **Any further comments?** - Well done

---------------------------------------------------------------------------------------------------------------------

**Name:** User E
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - When you search a description of the procedure so that the lay person understands what procedure they are selecting
4. **How likely would you use the website if it were relevant to you?** -4: Very likely

**Scenario 1**

1. **Write your exact search term below** - chest pain
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 313 - CHEST PAIN
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - Lake Pointe Medical Center, $4781

**Scenario 2**

1. **Write your exact search term below** - Concussion
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search -** Concussion
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price -**

**General Questions 2**

1. **How easy was it to navigate through the website?** - 4: Easy
2. **Which features did you like about our website?** - Well designed, nice colours
3. **What would you most like to see improved about this website? -** Bit buggy, map stopped working, price needs to be displayed on the search terms
4. **Anything you disliked about the website?** - No
5. **Any further comments? -**

-------------------------------------------------------------------------------------------------------------------

**Name:** User G
**Date:** 2nd October 2019
**Has read and signed the informed consent form:** Yes
**Has read the participant information sheet:** Yes

**General Questions 1**

1. **How visually appealing is the website?** - 4: Very appealing
2. **Does the website provide:** - 2: Right amount of information
3. **What further information could be provided?** - Information on the type of procedure, when searching for a procedure, many results come up, information on the differences between them would be helpful
4. **How likely would you use the website if it were relevant to you?** - 3: Somewhat likely

**Scenario 1**

1. **Write your exact search term below** - Chest pain
2. **Write the name of the top search suggestions that came up when you finished typing in your search** - 313 - CHEST PAIN
3. **Did your search run as expected?** - Yes
4. **If you said Yes to the above, write hospital you selected and for which price** - Lake Pointe Medical Centre, $4648

**Scenario 2**

1. **Write your exact search term below** - Head injury
2. **Write the name of the top search suggestions, if any, that came up when you finished typing in your search** - None, later tried term “concussion” which yielded results
3. **Did you manage to find a procedure to help you?** - No
4. **If you said Yes to the above, write hospital you selected and for which price -**

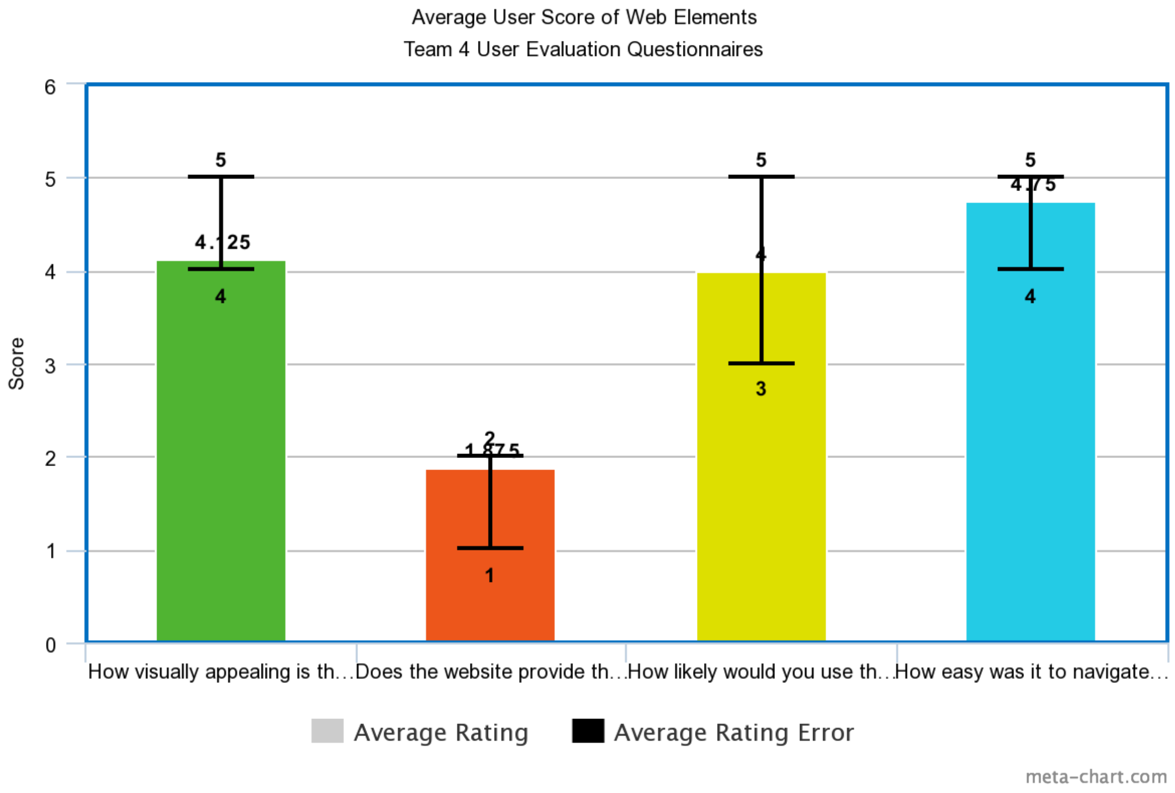
**General Questions 2**

1. **How easy was it to navigate through the website?** - 5: Very easy
2. **Which features did you like about our website?** - Simple layout, easy to use
3. **What would you most like to see improved about this website?** - After searching, remind the user of the search term. Provide results for more vague search terms, i.e. for scenario of the term “head injury”
4. **Anything you disliked about the website?** -
5. **Any further comments?** -

-----------------------------------------------------------------------------------------------------------------------

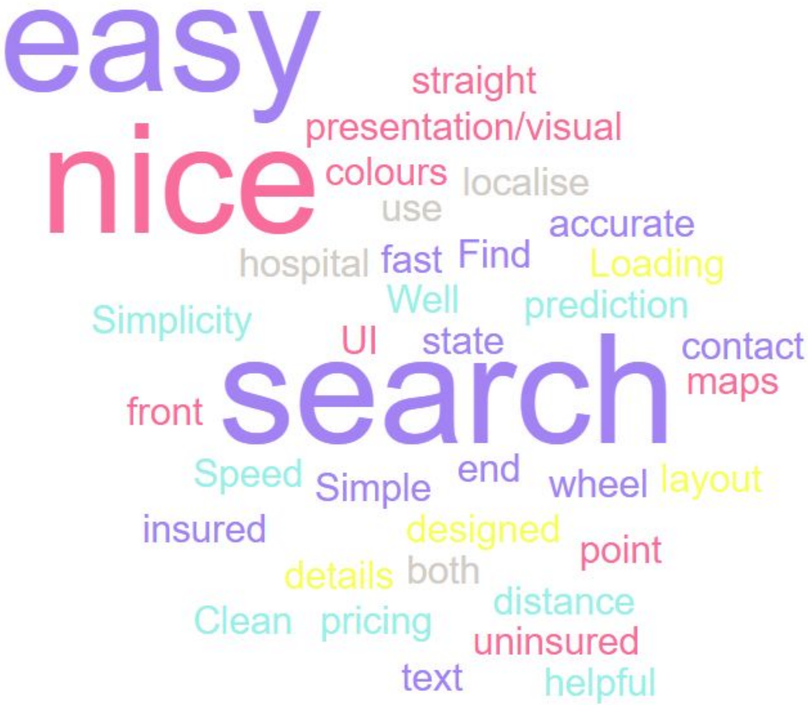
## Statistics

**Questionnaire Multiple Choice Scores (amount of information was rated out of 3, rest out of 5)**

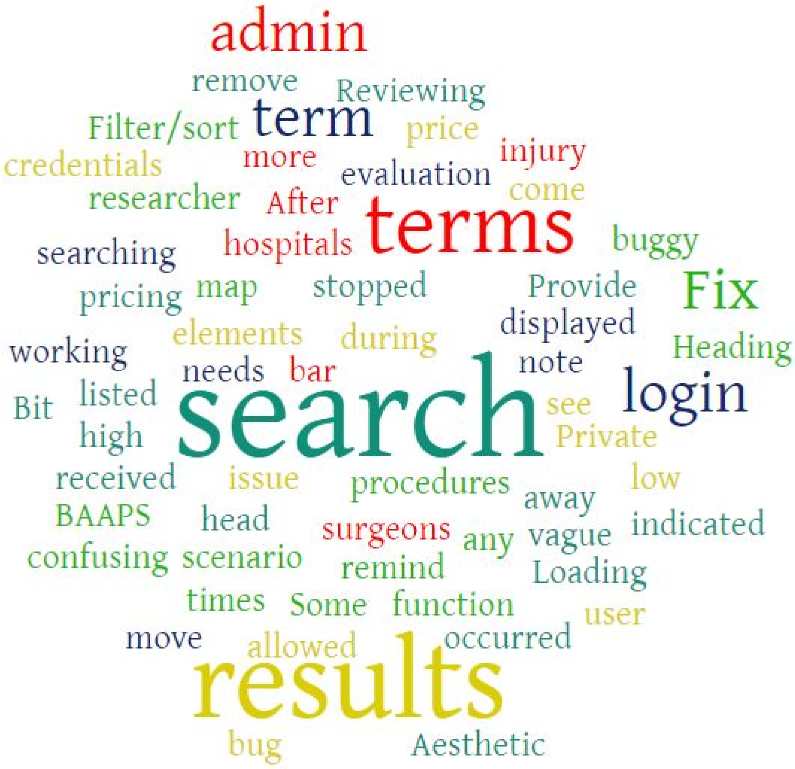


**Word Clouds for Question Answers**

**Which features did you like about our website?**



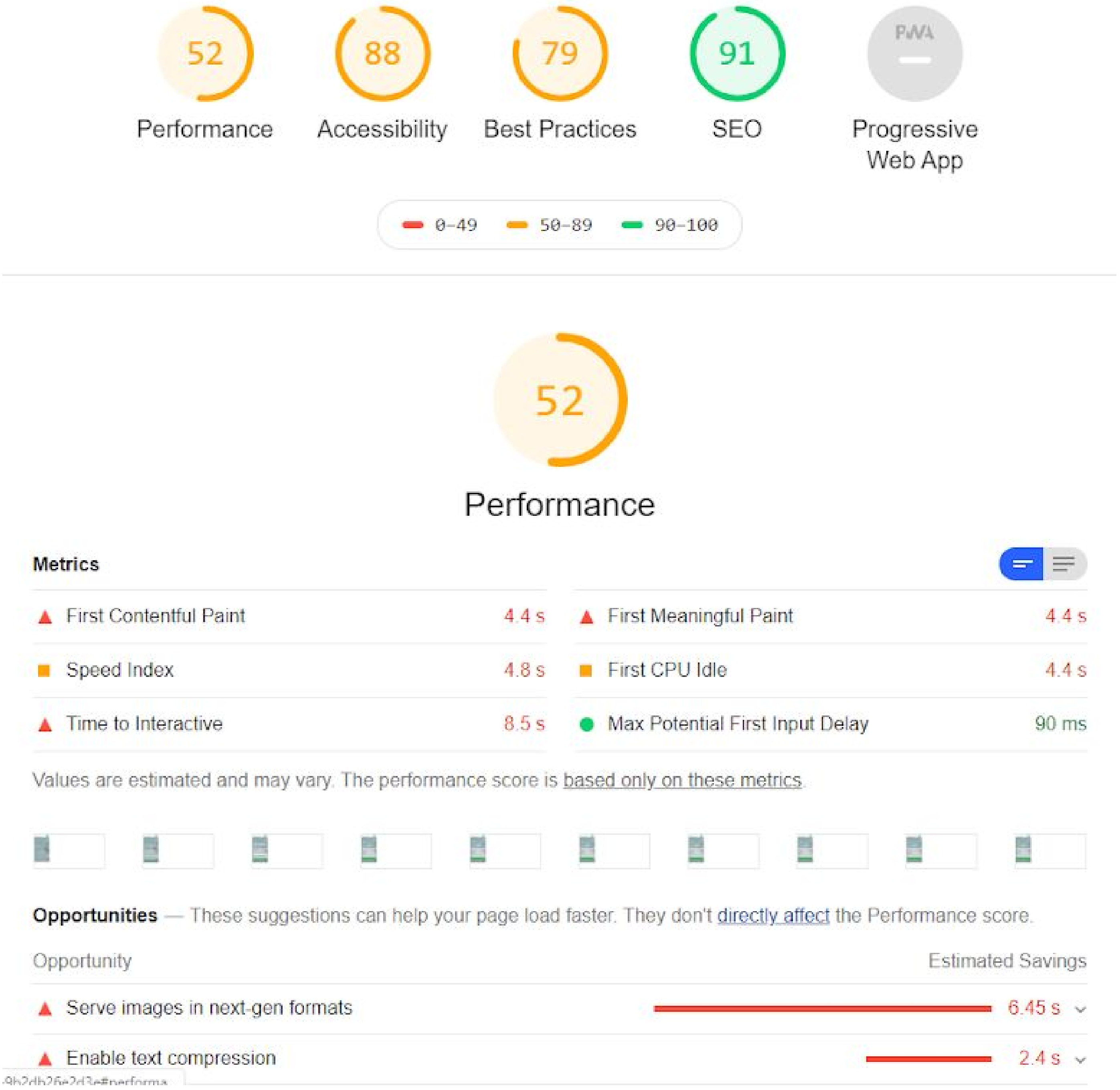
**What would you most like to see improved about this website?**



## Appendix H

**Title**: Lighthouse Accessibility Report

**Relevance**: Lighthouse is a Chrome extension that rates the accessibility of a website for the end user. An auto-generated report rated our website as 88% accessible for end users.



52 Performance

88 Accessibility

79 Best Practices

91 SEO

PWA Progressive Web App

🟥 0-49 🟧 50-89 🟩 90-100

52

Performance

**Metrics**

| | | | | | |
|---|---|---|---|---|---|
| ▲ | First Contentful Paint | 4.4 s | ▲ | First Meaningful Paint | 4.4 s |
| ■ | Speed Index | 4.8 s | ■ | First CPU Idle | 4.4 s |
| ▲ | Time to Interactive | 8.5 s | ● | Max Potential First Input Delay | 90 ms |

Values are estimated and may vary. The performance score is based only on these metrics.

**Opportunities** — These suggestions can help your page load faster. They don't directly affect the Performance score.

| Opportunity | Estimated Savings |
|---|---|
| ▲ Serve images in next-gen formats | 6.45 s |
| ▲ Enable text compression | 2.4 s |

9b2db26e2d3e#performa

## Appendix J

**Title**: User Evaluation Questionnaire

**Relevance**: A custom questionnaire we asked surveyees to fill out while performing an evaluation of our user search. The results of each questionnaire carried out can be found in Appendix G.

University of Dundee

# AC41004 Industrial Project
# Team 4
# User Evaluation Questionnaire

**This questionnaire will guide you through a series of test scenarios for interacting with the web portal of *CompareTheHealthcare.com.* Read the given scenarios and, for each one, answer the questions that follow by circling your answer.**

## General Questions 1

Have a go testing the web portal to get used to it and how it functions. Limit yourself to testing the search functionality only and try a few search queries based on procedures or illnesses you theoretically may have. Once you’ve done this, answer the following questions.

1. How visually appealing is the website?
   - Extremely appealing
   - Very appealing
   - Somewhat appealing
   - Not so appealing
   - Not at all appealing

2. Does the website provide:
   - Too much information
   - Right amount of information
   - Too little information

3. What further information do you think could be provided?

________________________________________________________________________

________________________________________________________________________

4. How likely would you use the website if it were relevant to you?

- Extremely likely
- Very likely
- Somewhat likely
- Not so likely
- Not at all likely

## Scenario 1

You are an American citizen living without medical insurance in the state of Texas (zip code 75189). After a hard shift at work, you've arrived home and found yourself suffering from some chest pain. Use the web portal to find a procedure that's right for you and fill out the questions below:

1. Write your exact search term below.

   ____________________________________________________

2. Write the name of the top search suggestions that came up when you finished typing in your search.

   ____________________________________________________

3. Did your search run as expected? **Yes/No**

4. If you said **Yes** to the above, write hospital you selected and for which price.

   ____________________________________________________

## Scenario 2

You are an American citizen insured by Medicare. Yesterday, you slept past your alarm and woke up realising you were late for work. In your haste to leave the house, you accidentally smack your head against your front door. Since then, you've been feeling faint and not as focused as usual. Use the web portal to find a procedure that could help you and fill out the questions below:

5. Write your exact search term below.

   ____________________________________________________

6. Write the name of the top search suggestions, if any, that came up when you finished typing in your search.

   ____________________________________________________

7. Did you manage to find a procedure to help you? **Yes/No**

8. If you said **Yes** to the above, write hospital you selected and for which price.

   _________________________________________________________________

## General Questions 2

Finally, please answer the last few questions.

4. How easy was it to navigate through the website?

- Very easy
- Easy
- Neither easy nor difficult
- Difficult
- Very difficult

5. Which features did you like about our website?

____________________________________________

____________________________________________

6. What would you most like to see improved about this website?

____________________________________________

____________________________________________

7. Anything you disliked about the website?

____________________________________________

____________________________________________

8. Any further comments?

____________________________________________

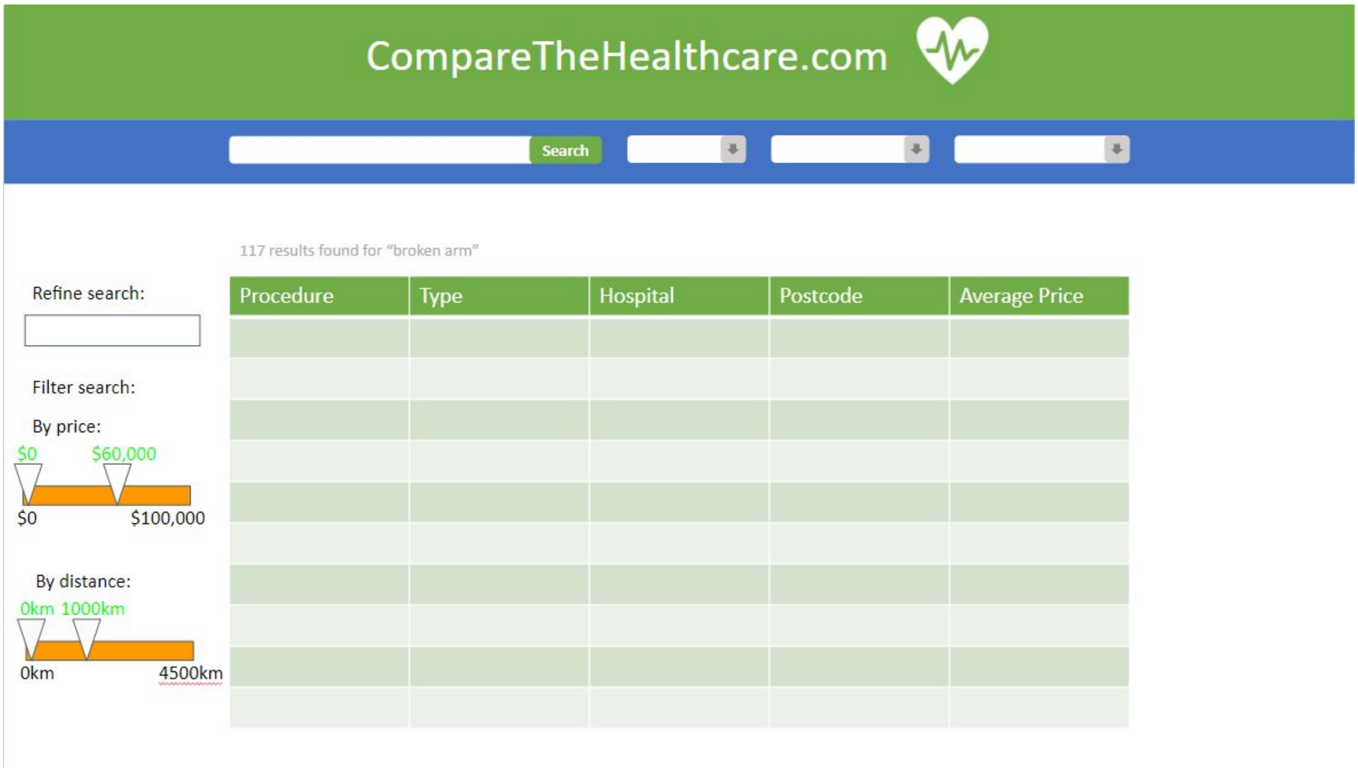
____________________________________________

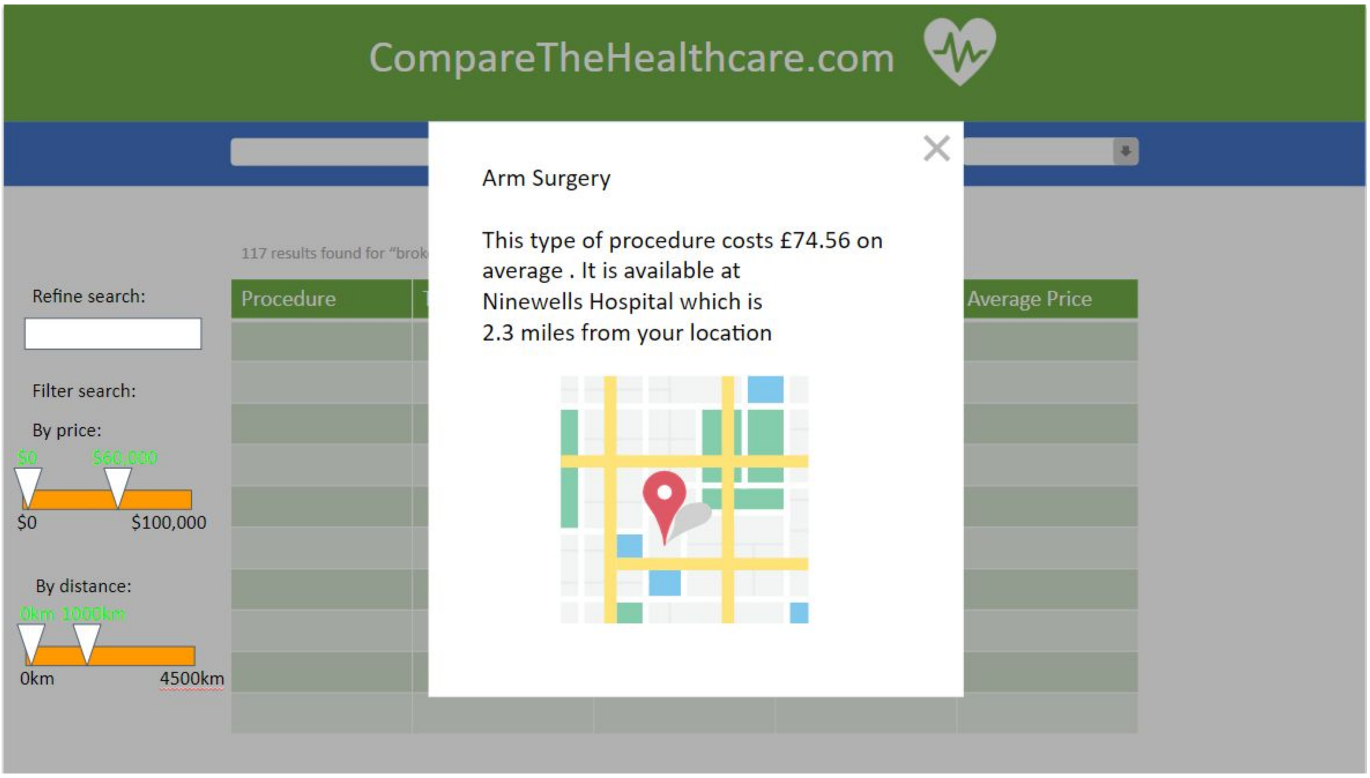
**QUESTIONNAIRE COMPLETE**
**THANK YOU FOR TAKING THE TIME TO FILL IT IN!**

# Appendix K

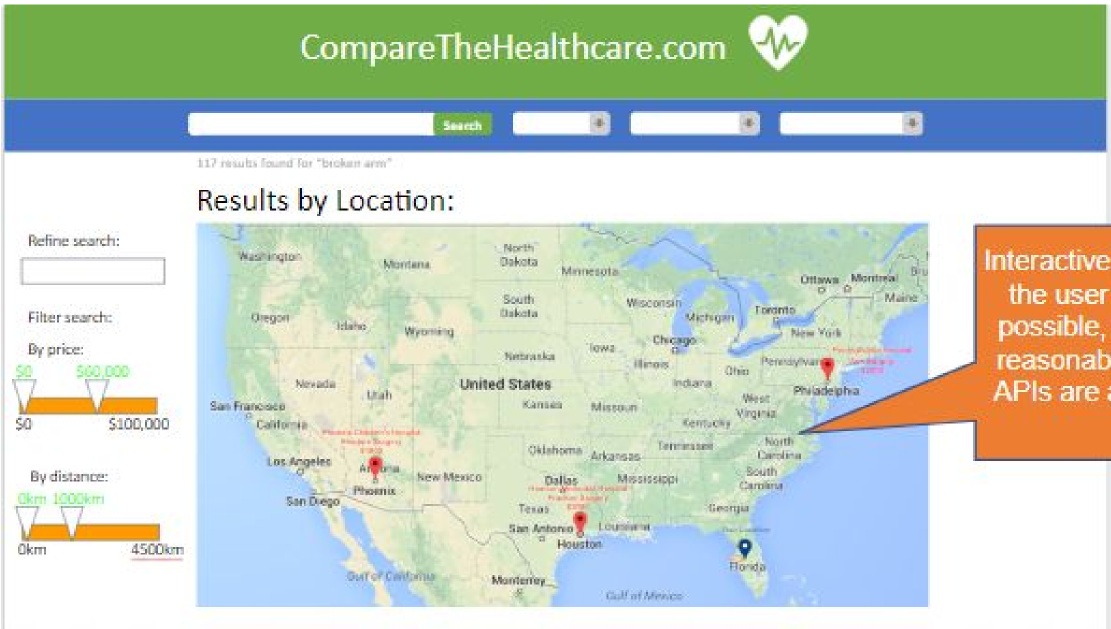
**Title**: User Interface Designs

**Relevance**: Initial design ideas for what the search pages would look like. This acts as a good comparison over how our ideas changed as time went on and our beliefs in what we could achieve in the time frame.





This type of procedure costs £74.56 on average . It is available at Ninewells Hospital which is 2.3 miles from your location



117 results found for "broken arm"

Results by Location:

Refine search:

Filter search:

By price:

$0 $60,000

$0 $100,000

By distance:

0km 1000km

0km 4500km

Interactive map. At least make allow the user to drag it around and, if possible, zoom in on the map to a reasonable degree. Not sure what APIs are available that freely allow this