InfoRent: Making Renters Content Since 2018

Sanja Dodos*
Computer Science and Psychology Major

Marcel Bernard†
Computer Science Major

Kevin Chen‡
Computer Science Major

Zane Li§
Computer Science Major

Abstract

The experience of browsing for rentals can be time consuming and stressful. InfoRent is a visualization tool designed to make that process more efficient and enjoyable. The tool allows users to visually compare rentals based on specific criteria such as location, price, number of bedrooms, number of bathrooms, pet allowance, and utilities. The tool will also help users with the average price across various neighbourhoods. This tool can be extremely beneficial to rental website providers such as Craigslist, UsedVictoria and Kijiji who all lack the ability to visually compare rentals and provide information about average neighbourhood prices.

1 Introduction

There is a need for renters in Victoria to have the ability to visually compare rental listings when searching for a place to live. There is also a need for renters to know the average rent prices in neighbourhoods across Victoria. Currently, there are some tools that allow users to identify rent prices for apartments in Victoria, but these tools neither reflect the differences in price for various neighborhoods nor do they reflect the prices for different types of dwellings such as townhouses, single bedrooms, or houses. The existing tools do not allow for easy comparison between rentals, which leads us to our research question: How can we spatially browse and compare rentals based on user criteria? For example, our application addresses the current shortcomings so that if a user searched for the rent cost of a townhouse near UVic, the application will display the cost of townhouses near UVic. Our application will show a customized map built using D3.js and Google Map API, with an overlay that displays rent details. There is a transparent choropeth map built on top of the map to show average neighbourhood prices. The map will allow user interaction such as zooming, filtering, panning and highlighting so that users can easily compare and identify prices. Our visualization tool will have a wide impact on rental website providers to allow their users to quickly and effectively search and compare rentals.

2 Related Work

Currently, there several similar tools available that we believe are not as efficient as they can be:

- PadMapper [10]: Allows users to search for rentals on a list/map and filter them based on price, length of stay, number of bedrooms, number of bathrooms and pet allowance but does not support a comparison feature. Although PadMapper provides a list of average prices for different apartment types, the prices are listed for an entire city. As a result, the prices are distorted as prices vary greatly from neighbourhood to neighbourhood within a city. For example, the apartment prices in James Bay, Victoria is much more expensive than Oaklands, Victoria, but based off of PadMapper, the user would not be able to tell the difference in price. Figure 1 shows how to browse rental listing on a map in Victoria. When a user selects a rental, the details and photos of the rental pop up on the side but does not show listings side by side.

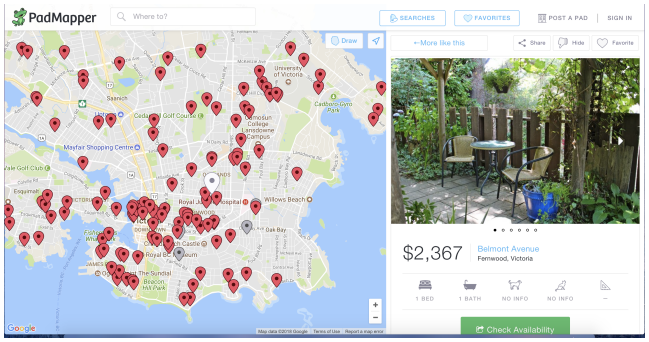

Figure 1: PadMapper

- Craigslist [3]: Allows users to search for rentals on a list/map and filter them based on price, length, number of bedrooms, number of bathrooms and pet allowance but lacks structure in listings and does not support a comparison feature. Craigslist also does not give an address for rental listings and it does not have a tool that lists the average expected rent price for different neighbourhoods in Victoria. Figure 2 shows how to browse rental listings in Victoria on a map in Victoria. A small icon comes up when a user selects a listing but does not give much information.

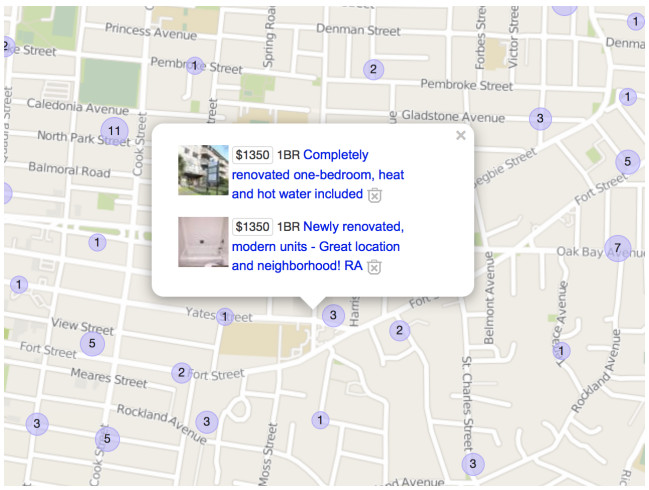

Figure 2: Craigslist

- UsedVictoria [15]: Allows users to search for rentals on a list/map and filter them based on price, length, number of bedrooms, number of bathrooms and pet allowance but lacks

*e-mail: sanjadod@uvic.ca
†e-mail: marcelb@uvic.ca
‡e-mail: wuhankein@gmail.com
§e-mail: zanelib1@gmail.com

structure in listings and does not support a comparison feature and does not give the average expected rent for different neighbourhoods in Victoria. Figure 3 shows how to browse listings in Victoria on a map. The icon that comes up shows some detail and photos regarding the listing however does not show more than one listing.

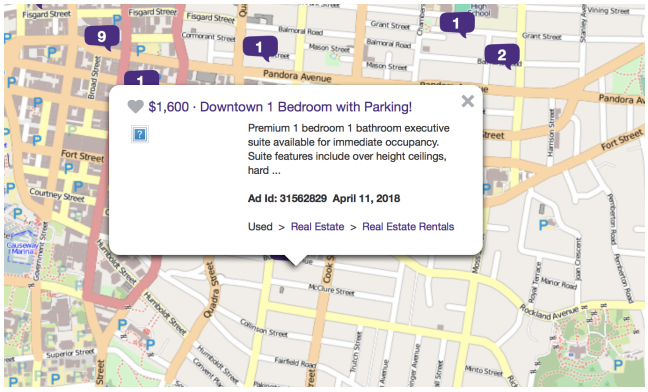

Figure 3: UsedVictoria

- Kijiji [7]: Allows users to search for rentals through specific real estate categories (eg. entire house, roommate, 1 bedroom, 2 bedroom etc.) and can filter within those categories based on price, number of bathrooms, pet allowance but does not support a map feature or comparison feature or average neighbourhood pricing. There is no way to browse listings on a map, however you can see the rental listing on a map once you select the listing but you cannot see other listings on the map.

- RentBoard Canada [11]: Allows users to search for rentals on a list/map based on price, number of bedrooms, number of bathrooms, pet allowance but does not support a comparison feature and does not give average price of the neighbourhood. Figure 4 shows what comes up once you select a listing. The pop up takes up the entire user's screen so you cannot see the location anymore or compare to other listings.

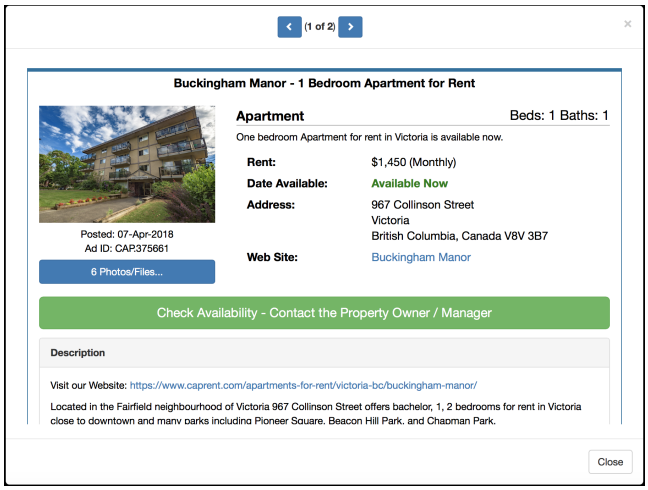

Figure 4: RentBoard

Overall, each of these tools uses their own visualization techniques to display the same information. They all lack two things: the ability to visually compare rentals including photos and details as well as not showing the average price of neighbourhoods.

3 Approach

Our approach to the visualization design will be discussed with respect to the what-why-how framework discussed in the course text [16].

3.1 What

The types of data were identified early on in order to determine what data abstractions would be appropriate. There were two primary forms of data that needed to be shown, rental listings and Victoria neighbourhood boundaries. Rental listings consisted of both categorical binary attributes, such as pets/no pets, and ordinal attributes such as price. These combinations of attributes and attribute types meant that the rental posting data lent itself to a table representation. In addition, rental postings also contain location data, which can be represented most naturally by a geometric position on a map. The second type of data, neighbourhood boundaries, also made sense to represent as geometric regions on a map.

There were two sets of derived attributes obtained from the original data. The first of these derived attributes was the average rental prices for each neighbourhood region. This data was derived by averaging the rental prices of each rental posting within a neighbourhood (normalized for number of bedrooms). The second derived attribute was the difference in average price and actual price for each rental unit. This essentially transforms the price scale from sequential (starting at 0 and increasing), to a diverging scale centred on each neighbourhoods average price. This derived data allowed for showing the difference in price of a rental compared to the average prices in its neighbourhood, either above or below.

3.2 Why

The abstract tasks that the visualization would support were determined with rental browsing users in mind. The primary tasks identified include browsing for rentals, including location and attributes, and comparing the location and attributes of a set of rentals. Some light analysis tasks may also be desirable for users interested in using the derived average rental prices per neighbourhood.

3.3 How

The data was represented using two primary visual encoding idioms, a map view and a tabular comparison view.

3.3.1 Map View

Two different map views were tried before arriving at the final view implementation. The first view considered was primarily a choropleth map view, as shown in figure 12. This view effectively displayed the average rental prices, however the strong colours were somewhat prohibitive to viewing the locations of individual rentals (pictured in figure 12 as small blue dots). The next iteration moved the primary view to a less distracting geographic map display that used the choropleth map as the minimap portion of a birds-eye map (see figure 5). This had the advantage of separating the average rental price information from the rentals themselves, however it made it more difficult for users to link between the minimap and actual map view.

The final implementation (see figure 6) took a hybrid of these two approaches. First, the minimap view was removed and the choropleth map was layered on top of the geographic map view. The choropleth layer was made transparent by default and allowed for the original colouring to be uncovered via mouse-over. This approach offered the advantage of having the rental positions and average prices located within the same space, and offered flexibility in terms of how much or how little average price information a user may want to see.

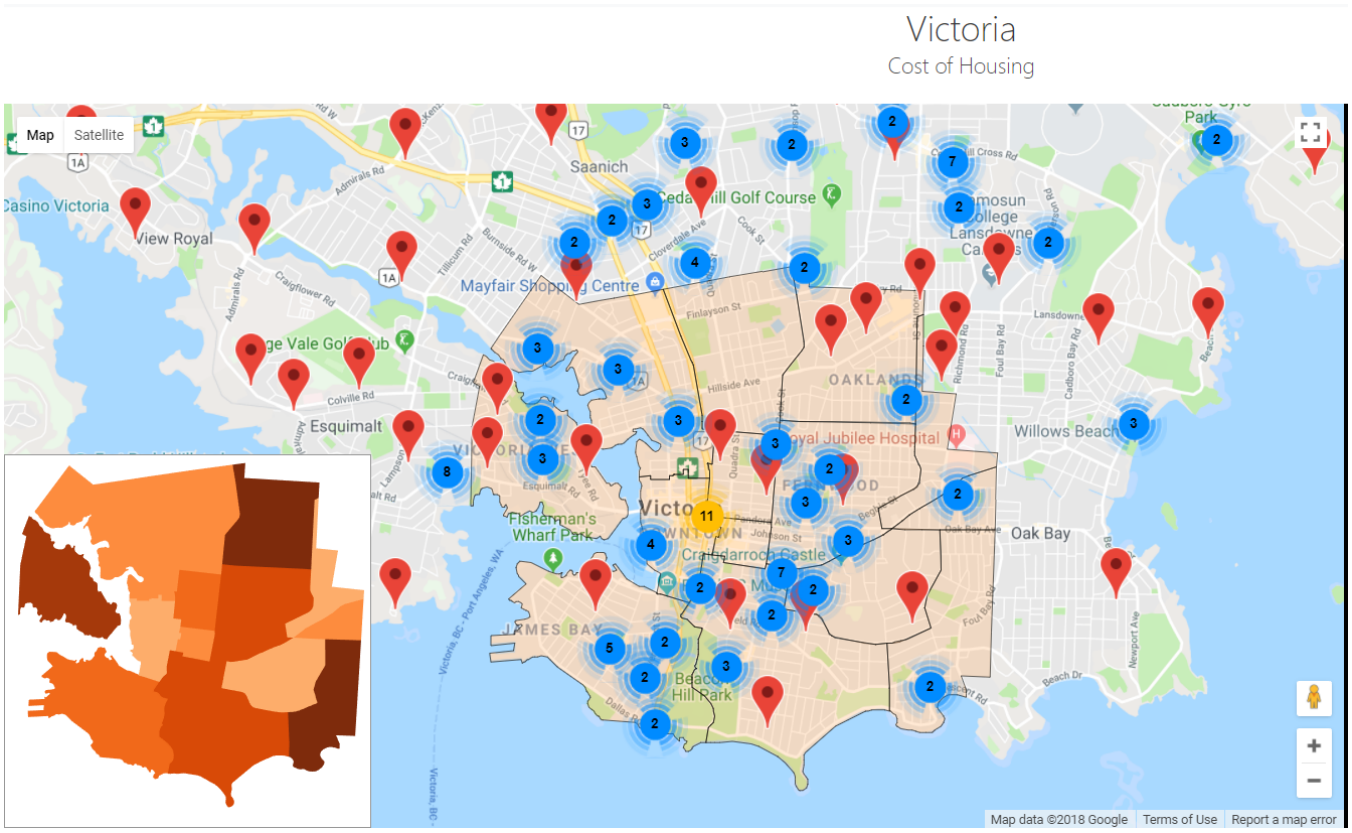

Figure 5: Previous view displaying minimap view support.

Several modes of interaction are supported for the final geographic map view. The map allows for geometric zooming and panning and it also supports semantic zooming for rental points (see figures 7 and 8). As a user zooms out, points are dynamically aggregated into single points with counts of the number of points contained within them, and as the user zooms back in these aggregated points are expanded into individual rental positions. The semantic zooming provides the advantage of reducing visual clutter as the user zooms out and the map size becomes smaller. In addition to panning and zooming, individual rentals can also be selected, which will result in the expansion of a window containing pictures of the selected rental, and information regarding its attributes.

The choropleth map made use of a single shade of orange, which should be easier for colour blind users to distinguish from the blue shades present on much of the rest of the map view. Luminance was varied in order to indicate differences in average rental prices, with higher luminance indicating lower average rental prices, and lower luminance indicating higher average rental prices. This offers an advantage to colourblind users, and will work effectively even on a black and white display.

3.3.2 Comparison View

The comparison view, shown in figure 9, is opened separately, once a user has selected rentals for comparison. This view is implemented using a table, which has several advantages. A table view is simple and easily understood by most users, and it also allows the display of a mix of attribute types, which is important because the data itself contains a mix of both categorical and ordinal attributes. The disadvantage with tables is that they can quickly become difficult to extract information from as the number of cells increases, however in this case users are limited to only comparing up to six rentals at a time, which is a manageable size for this view.

An earlier implementation of this view, shown in figure 10, moved the rental prices into a bar chart view above the comparison table. This had the advantage of making it easier to compare the relative magnitudes of rental prices, however it used a considerable amount of screen space, and it wasnt clear that it was much easier to read than simply having the prices in the table view itself. The bar chart was removed in the final implementation and it was instead replaced with a direct comparison to average neighbourhood prices below each price in the table. This comparison value shows how much a rentals price differs from the average price within its neighbourhood, using green with a downward facing arrow to indicate that a rental is below the neighbourhood average, and red with an upward facing arrow to indicate that it is above the average price. Green and red are commonly understood to indicate positive and negative differences respectively, however this information is also redundantly encoded with arrows, since the green and red colours may not be distinguishable for colourblind users.

4 Implementation and Technology

The project is implemented as a web application using a number of different technologies.

4.1 Backend Technology

4.1.1 Server Technology

The web server runs from an Amazon AWS EC2 instance [1]. Initially, Amazon AWS was challenging for us because we did not have prior experience and the system is quite complicated compared to other web hosting services we have used in the past such as Digital Ocean. Since we felt that knowing how to work with AWS would be a desirable skill, we decided to learn how to use it. Eventually,

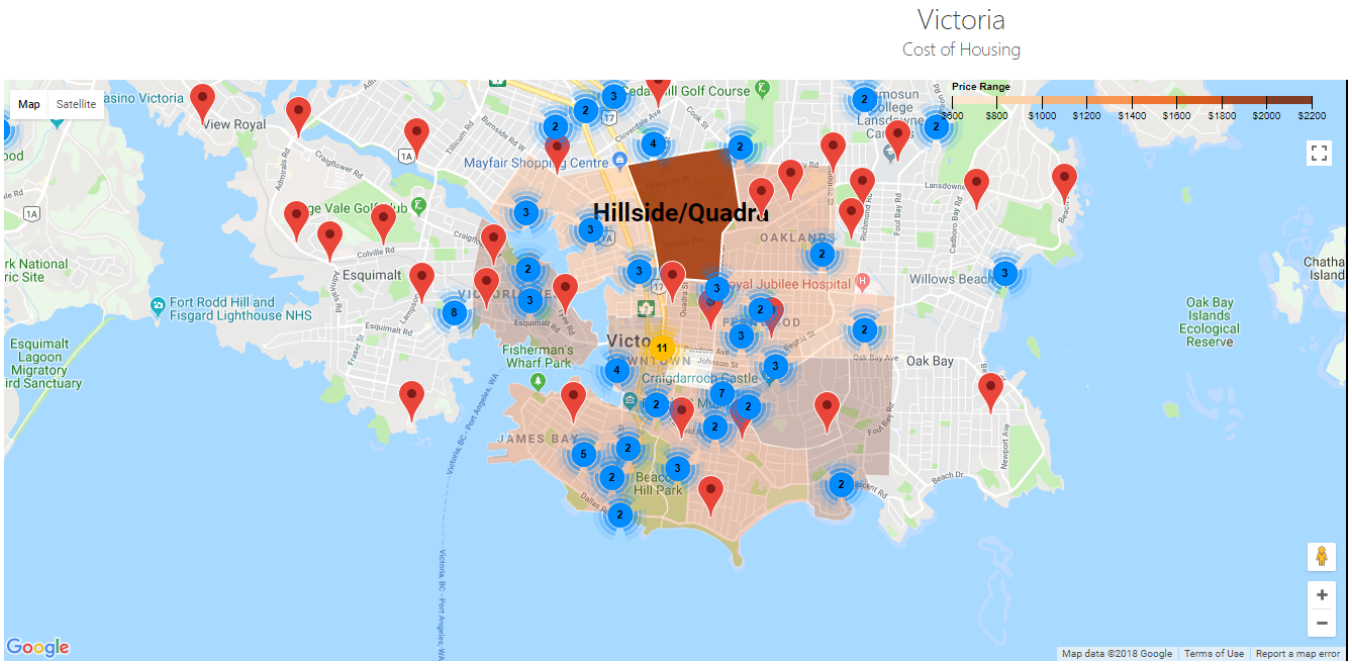

Figure 6: Final implementation of the geographic map view.

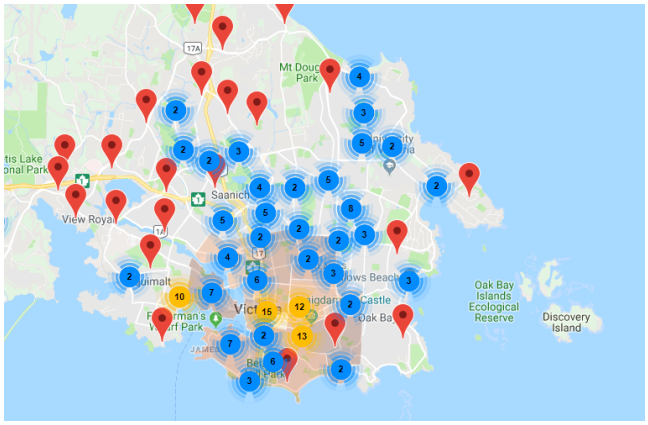

Figure 7: Semantic zooming example showing a zoomed-in view of rental points. Note that the circular red aggregation marks have been expanded into blue and yellow aggregation marks, along with individual points (shown as red marks with black circles inside).

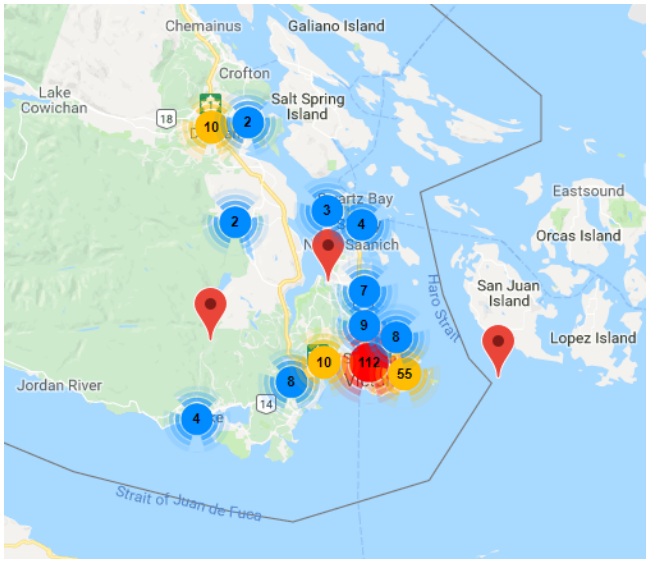

Figure 8: Semantic zooming example showing a zoomed-out view of rental points. The aggregation marks are show in red, blue, and yellow.

we became familiar with AWS and now we feel it is a reliable web hosting solution. The backend is relatively simple and is implemented via the Python Flask web framework [5] due to its ease of use. We considered building the backend with other technologies such as Node.js, but we eventually balked at the idea because we did not have prior experience working with Node and we did not feel comfortable learning a completely new backend framework. Many of our group members had very busy schedules this semester, and we decided that the safe option would be to build a backend with a technology we had prior experience.

4.1.2 Rental Data Gathering and Preparation

In order to gather the necessary data for our project, we had to scrape rental data from various rental agencies. Data scraping was carried out somewhat successfully on Craigslist using the Python scrapy library [12]. Although we meticulously scraped from Craigslist so that they would not detect us, Craigslist eventually discovered our scraping and blocked us from further scraping them. As a result, we acquired 200 rental records from Craigslist. On the other hand, scraping from Kijiji was successfully carried out using the kijiji-scraper Node.js package. The kijiji-scraper package made scraping from Kijiji a simple and swift process [8]. However, Kijiji does not possess as many rental records as Craigslist and the data was unstructured. Nonetheless, we acquired over 100 rental records from Kijiji. Subsequently, we combined the rental data from Craigslist and Kijiji into a single file. In cases where fields were missing, we

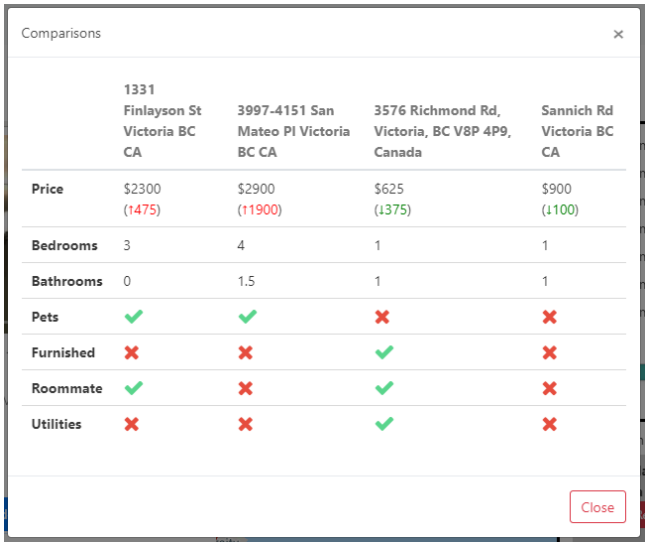

Figure 9: Final comparison view implementation

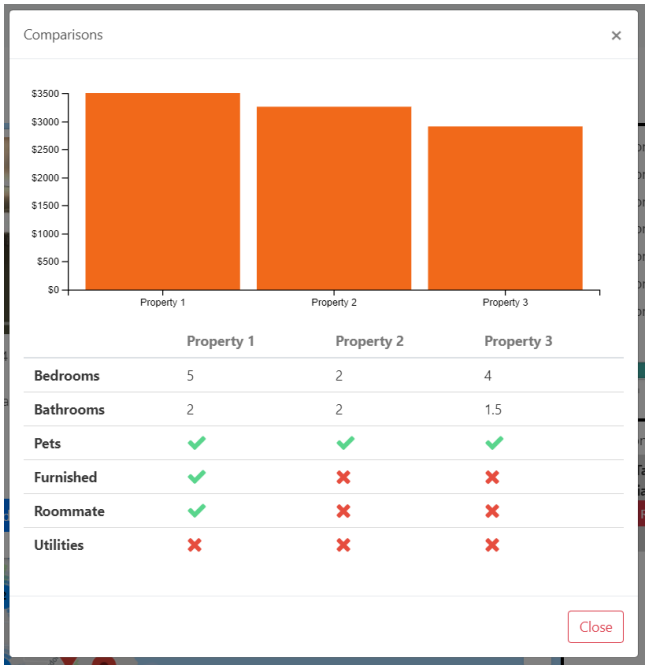

Figure 10: Previous comparison view displaying price bar charts.

filled those with arbitrary values.

4.1.3 Neighbourhood Data Gathering and Preparation

Gathering neighbourhood boundary data turned out to be much more difficult that we anticipated as the vast majority of cities in Greater Victoria do not release neighbourhood boundary information. Other than Victoria, not a single other city in Greater Victoria provided boundary information. Municipalities such as Saanich have a lot of geography data and even provides an interactive map that allows people to explore the city topology and geography. However, these municipalities did not publicly provide any means of downloading neighbourhood boundary data. As a result, we could only acquire the neighbourhood data for the City of Victoria in shp format. At the time, we decided that we would move forward even though we only had the boundary data for Victoria. Our plan was to continue working on the project while searching for the boundaries for other cities in Victoria. If we can find the data, then we can incorporate it to our project, but it was not feasible for us to continue waiting for the data without moving forward with the project. Once we acquired both the rental dataset and neighbourhood boundary dataset for Victoria, some data preprocessing was done using Python, such as determining which neighbourhoods each rental was contained in, while data formats from Kijiji and Craigslist were standardized and consolidated manually into a single CSV file and the boundary dataset was converted to geoJSON.

4.2 Front End Technology

A combination of tools were used to design the overall look of the visualization tool:

- HTML, CSS, and Javascript: Overall web page visualization design.
- Bootstrap 4 [2]: Overall web content style design.
- GeoJSON and TopoJSON [14]: Map and neighbourhood geography data and the locations of individual rental properties are extracted from a common CSV file.
- D3 Javascript library [4]: Originally used on creating choropleth map and comparison feature. For the latest iteration of the visualization tool, we took the advantage of D3's unique manipulation on DOM(document object model) elements and its powerful visualization components to build price legend, picture slideshow, comparison, etc.
- Google Map API [9]: Customed Birds Eye View Map.

Initially, we used HTML, CSS, Bootstrap, and Javascript to build a standard index page for our visualization. To show our pricing map we used a combination of D3.js and Javascript; these technologies allowed us to display a choropleth map with a colormap legend attached above it. Later, we used Google Maps to display all of rental properties as we used Google Maps markers to display each rental. We were surprised to learn that Google Maps can also be used in collaboration with D3.js. As a result, we added a coloured overlay atop Google Maps using D3.js so that average rental prices for each neighbourhood is displayed to users. The combination of these front end technologies allowed us to not only show each rental property on a map, but also illustrate the price context for the surrounding areas. Furthermore, Javascript allowed us to include a filter feature so that users can filter properties depending on a specific criteria.

4.3 Challenges

Initially, the Google Maps Javascript API [9] was used for the initial implementation of the map visualization, however due to difficulty using the API, and a lack of knowledge of the internal workings (the code is closed source), it was replaced in favour of the D3 Javascript library. Furthermore Google Maps did not allow us to easily convert GeoJSON polygons into Google Maps polygons. As a result, it was difficult to translate the polygon objects into modifiable objects that allowed us to run detailed queries on the them. However, Google Maps allowed easy previews of neighbourhood boundaries. We could rapidly view boundaries by importing kml files into Google Earth [6] and then waiting momentarily before the resulting boundary lines appeared in Google Earth. When we eventually acquired the neighbourhood boundaries for Victoria, we first imported the data to Google Earth so that we could verify that the boundary lines were correct. [13] was used to convert shp files to kml files.

Although D3.js was easier to work with than Google Maps in terms of creating a choropleth map that shows average rent pricing

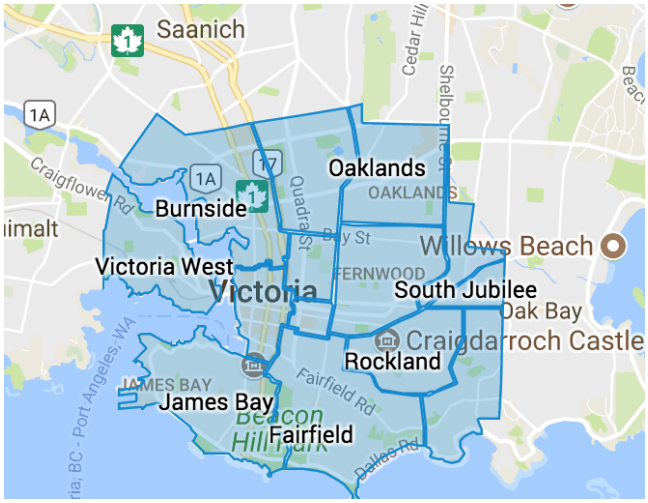

Figure 11: Dashboard original sketch for comparison feature

for a neighbourhood, adding additional functionality such as points on the map was rather difficult. Since one of the key bits of information seeked when finding rentals is knowing where the property is located, visualizing the property with map view (with streets) allows users to relate the property geographically. Although D3.js allows us to show where rental properties are located on the choropleth map, users were not able to find the specific or surrounding roads for rentals. As a result, we realized that if were to continue with D3.js, our project would have a significant flaw.

Therefore, we added Google Maps back to project. Using Javascript with Google Maps allowed us effectively display where each rental property is located and users are given the opportunity to trace the surrounding streets to see the accessibility of the rental. Additionally, we kept D3.js in our project because it is an simplistic method to display a colormap overlay (based on average rental prices for each Victoria neighbourhood region) atop the Google Maps) as well as a colormap legend at the top of Google Map.

5 Findings

Our tool was evaluated through use case studies and self evaluation.

5.1 Usability Testing

User studies were carried out by student volunteers for usability testing. Two main tasks were carried out by the users. The first task was to find an ideal one bedroom/one bathroom apartment in Oak Bay which allows pets, includes utilities and is below average market price and then do the same but with another rental website provider mentioned in the related works such as Craigslist and UsedVic. The results are shown in Table 1.

Table 1: User Studies for Task 1

Student	InfoRent	Craiglist	Time Difference
Student 1	35 sec	115 sec	80 sec
Student 2	40 sec	100 sec	60 sec
Student 3	37 sec	120 sec	83 sec
Student 4	45 sec	130 sec	85 sec
Student 5	50 sec	125 sec	75 sec

To summarize the above data, students were able to complete the task three times faster on average through InfoRent compared to the other website rental provider. The users were also not able to find the average neighbourhood pricing on the other website.

To further validate this observation, we conducted a second task which was the same as the first task but with different and more complex criteria: find a 3 bedroom/2 bathroom house that is pet friendly and may or may not include utilities in the neighbourhood of their choice but below market value. The results are summarized in Table 2.

Table 2: User Studies for Task 2

Student	InfoRent	Craiglist	Time Difference
Student 1	120 sec	370 sec	250 sec
Student 2	130 sec	385 sec	255 sec
Student 3	145 sec	400 sec	255 sec
Student 4	125 sec	350 sec	225 sec
Student 5	137 sec	410 sec	273 sec

The task took longer for both InfoRent and Craigslist compared to task 1 but was on average three times faster on InfoRent.

Furthermore, we allowed users to play with the tool to define their own tasks and provide feedback if they incurred an error or got confused. The feedback included:

- Interactivity was intuitive and user friendly
- Missing side by side photo comparison
- Average prices for neighbourhoods is helpful
- Filtering could be a sliding scale
- Expanding neighbourhood boundaries

5.2 Team Self Evaluation

Throughout the project development, each team member independently reviewed the features as they were implemented in order to improve the quality and effectiveness of the tool. This was done because each member worked independently on features and required a fresh set of eyes for feedback on progress. Overall the final tool was collectively agreed upon for satisfying all of our requirements.

6 Discussion

Some strengths for our tool include visual comparison ability, interactivity and good usability.

We also recognize some weaknesses in our tool including:

- No list view (compared to Craigslist, etc.) which makes it more difficult to search exhaustively through all of the rentals.
- Difficulty in getting accurate data without missing information.

The biggest lesson we have learned is that web development is hard, time consuming, and tedious.

For the future work of our visualization tool, as this practical tool serves for one purpose which is to help user spatially find optimal rental places, our tool will only gather data from other websites like Kijiji, Craigslist, etc. Therefore we are thinking to include a link to the original website where it is posted to the rental information window. Also the current choropleth map for average neighbourhood price will be expanded and eventually covers the whole of Greater Victoria. We would like to see more features within the tool including adding photos to the comparison dashboard for easier viewing. It would also be useful to adapt the tool into a mobile application for easier access and mobility.

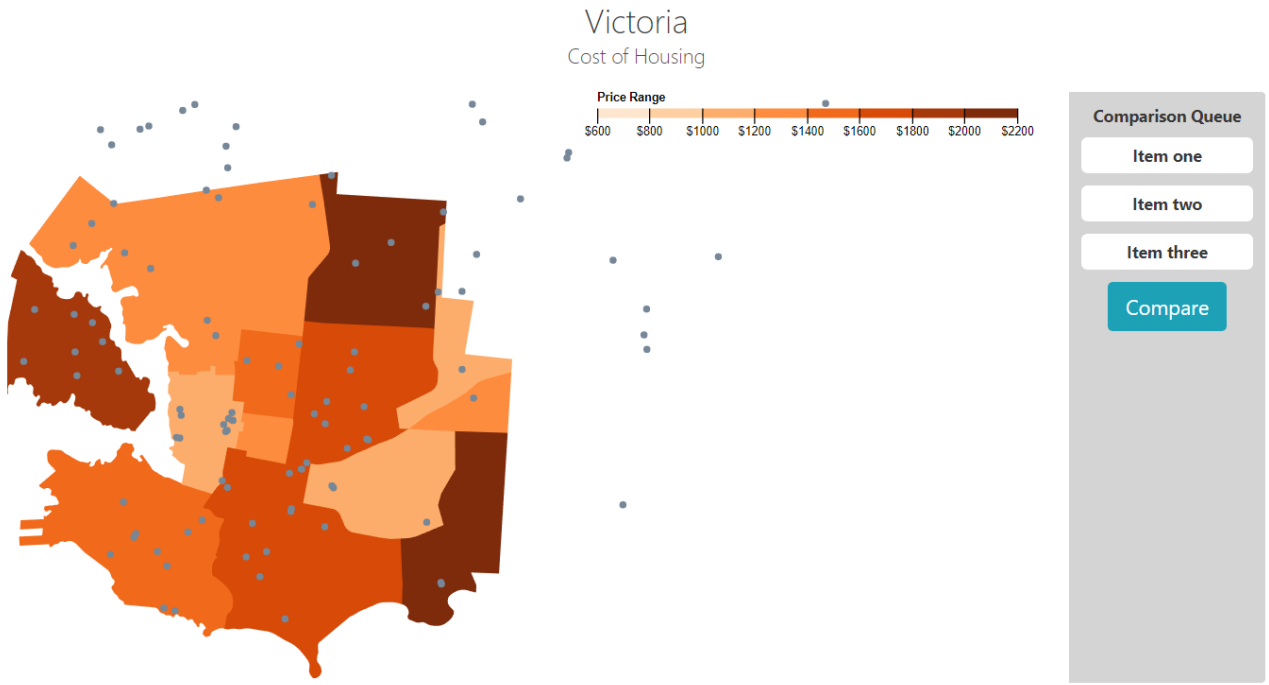

Figure 12: Previous iteration of the visualization, showing the original choropleth map view with small blue marks representing rentals.

7 Conclusion

In conclusion, current rental website providers such as Craigslist and UsedVic lack the ability to visually compare rental listing and view average neighbourhood pricing. Our visualization tool, InfoRent uses several key information visualization principles such as a choropleth map and dashboard to address these issues. With the further addition of features, this tool can be used by website rental providers to increase the effectiveness of their services.

Acknowledgments

The authors wish to thank Dr. Peggy Storey, Omar Elazhary, and David Johnson for their helpful mentoring throughout this project. We would also like to thank our classmates for helping us with our tool evaluations.

References

[1] Amazon ec2. https://aws.amazon.com/ec2/. Accessed: 2018-03-04.
[2] Bootstrap. https://getbootstrap.com/. Accessed: 2018-04-13.
[3] Craigslist. https://victoria.craigslist.ca/d/apts-housing-for-rent/search/apa. Accessed: 2018-03-04.
[4] D3: Data-driven documents. https://d3js.org/. Accessed: 2018-03-04.
[5] Flask: A python microframework. http://flask.pocoo.org/. Accessed: 2018-03-04.
[6] Google earth. https://earth.google.com/web. Accessed: 2018-03-04.
[7] Kijiji. https://www.kijiji.ca/b-real-estate/victoria-bc/c34l1700173. Accessed: 2018-03-04.
[8] kijiji-scraper: A lightweight node.js for retrieving and scraping ads from kijiji. https://github.com/mwpenny/kijiji-scraper. Accessed: 2018-03-04.
[9] Maps javascript api
. https://developers.google.com/maps/documentation/javascript/. Accessed: 2018-03-04.
[10] Padmapper. https://www.padmapper.com. Accessed: 2018-03-04.
[11] Rentboard canada. https://www.rentboard.ca/rentals/index.aspx. Accessed: 2018-03-04.
[12] Scrapy: A fast and powerful scraping and web crawling framework. https://scrapy.org/. Accessed: 2018-03-04.
[13] Shp to kmz converter online. https://mygeodata.cloud/converter/shp-to-kmz. Accessed: 2018-03-04.
[14] Topojson. https://github.com/topojson/topojson/wiki. Accessed: 2018-03-04.
[15] Usedvictoria. https://www.usedvictoria.com/classifieds/real-estate-rentals. Accessed: 2018-03-04.
[16] T. Munzner. *Visualization Analysis & Design*. CRC Press, Boca Raton, Florida, 2014.

8 APPENDIX

8.1 Figures

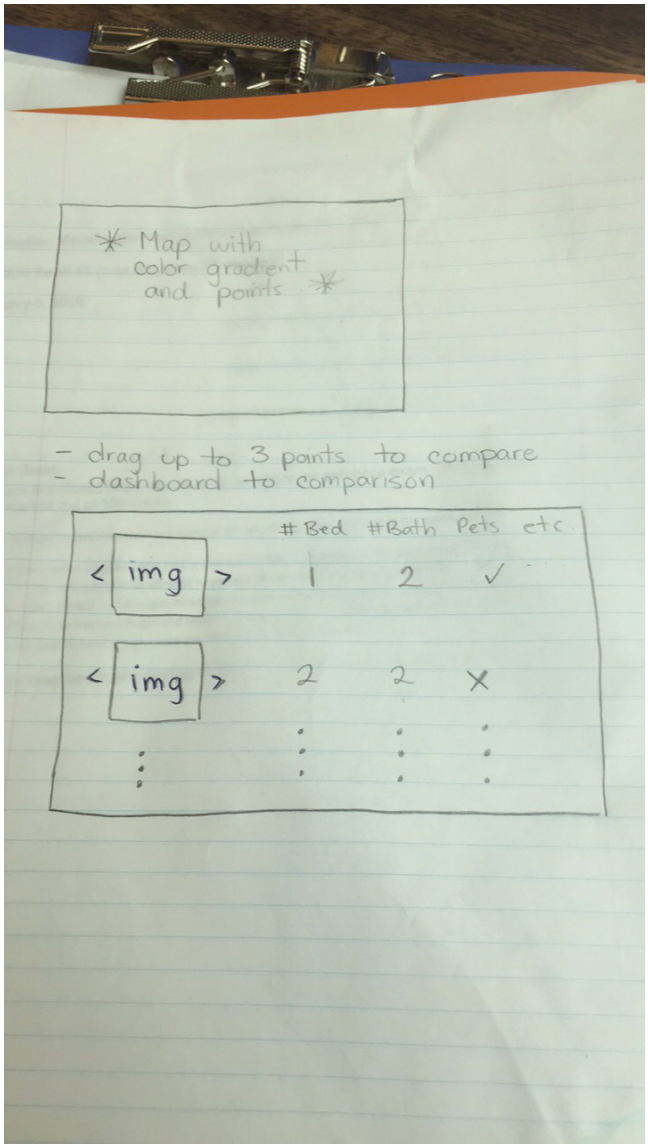

Figure 13: Dashboard original sketch for comparison feature

8.2 Collaboration Techniques

Contribution and collaboration by group members was done through three main platforms: GitHub: Used extensively by the group to store code and keep track of version control. Also it is used by the mentors to observe and give feedback on the progress of our project. Google Drive: Used to store data and write reports. We selected Google Drive because its tools such as Google Docs allows group members to collaborate in parallel. Slack: Used to exchange and discuss opinions and ideas. Also used to schedule meetings and upload relevant files.

8.3 Milestones

Table 3: Milestone table

Milestone	Assignment	Due Date
Background Research	Everyone	Feb 4
Web Server	Marcel	Feb 4
Kijiji Data	Zane	Feb 11
Craigslist Data	Marcel	Feb 11
UsedVic Data	Sanja	Feb 11
Prepare Data	Kevin	Feb 15
Map layout	Kevin	Feb 16
Choose front end framework	Zane	Feb 18
D3 prototype (just locations)	Everyone	Feb 22
UI design	Sanja	Feb 24
Interactivity	Kevin and Zane	March 8
Rental Points	Marcel and Kevin	March 11
Filters	Zane and Marcel	March 13
Comparison Features	Everyone	March 19
Visualization Assessment	Sanja	March 25
User Case Studies	Sanja	March 30
Feedback Integration	Everyone	April 3
Final Presentation	Everyone	April 5
Final Report	Everyone	April 12