

# **USA City Search**

## **CS 5934 Capstone Project: Final Product Submission**

### **Team members:**

Shreya Malhotra  
Kenneth Powell  
Joseph Ding  
Pallavi Padmanabha  
Yingfeng Fang

Under The guidance of  
Dr Sally Hamouda

## 1. Description

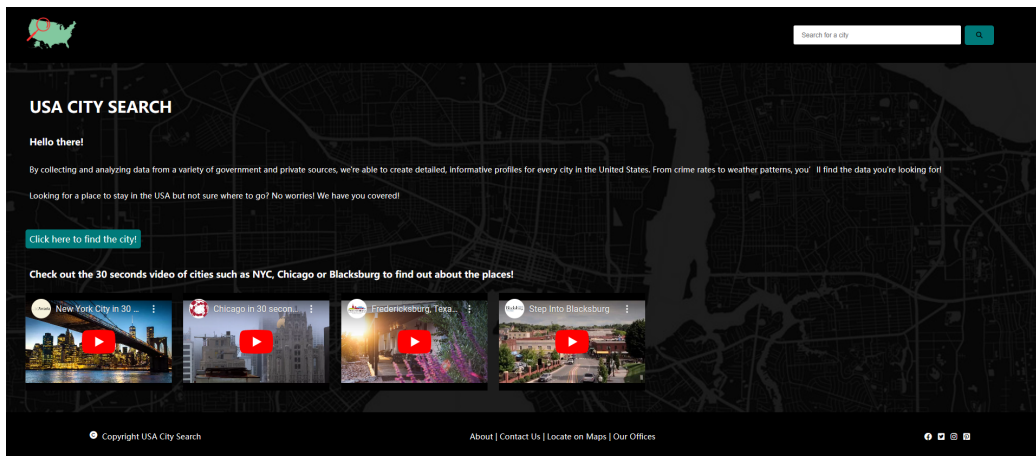
Imagine you are a young professional, you recently got a new job, and your lease is ending. The job is remote and you can choose to live in any location within the US you want. However the current city you are staying at is on the east coast with a high crime rate and a high state income tax. You are planning to move to a city with a lower crime rate and a lower state income tax which has a beach you can go to. Since you also speak Spanish you would want to be where other people speak Spanish so you can have people to practice with, and you would also love to have a city and beach vibe. Any places you can think of? You can either go ahead with Texas or Florida as those suit the best description. If you wish to compare cities, our website will provide you with all this information and with easy accessibility where you can sort and rank results too.

## 2. Product functionalities

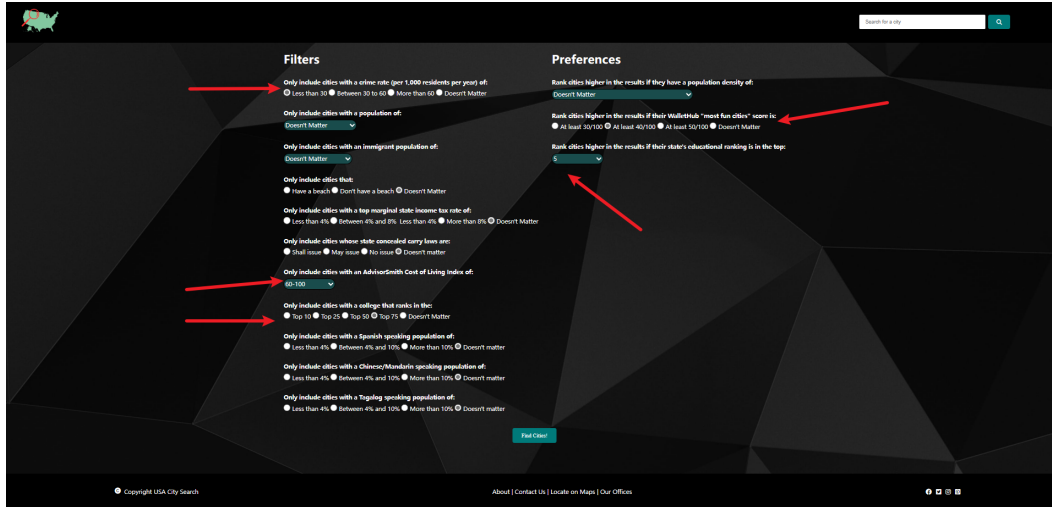
### 2.1 Features Overview

Suppose I am a new user who just started using the product.

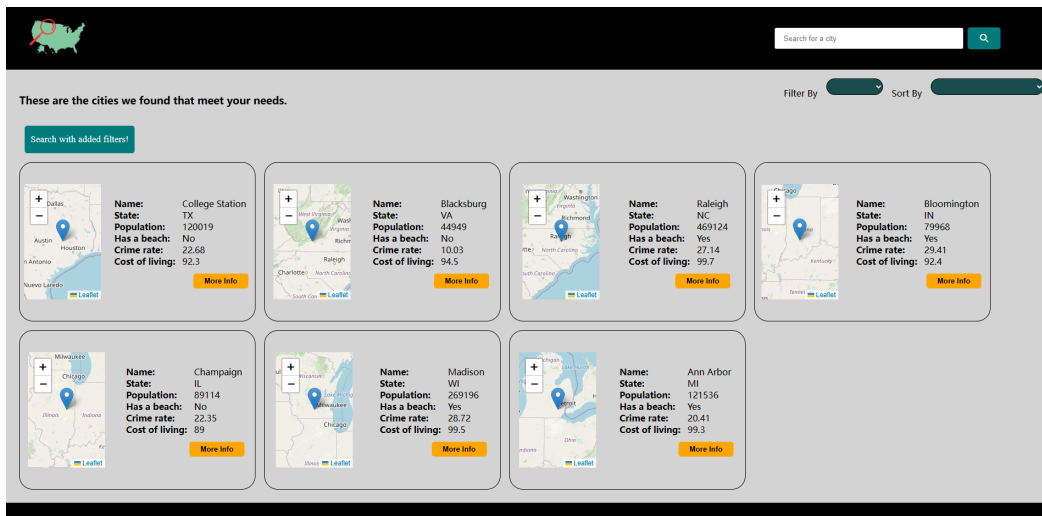
First, I will land on the landing page.



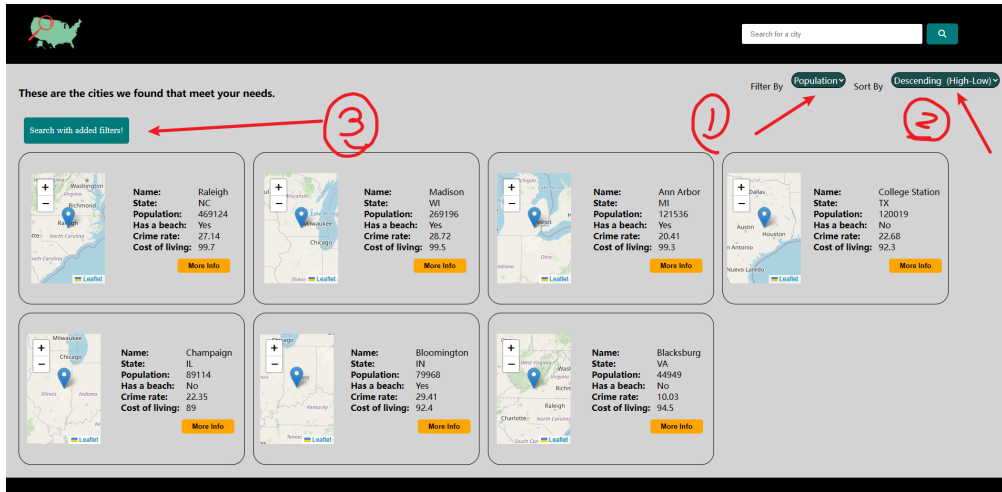
From here, I can check some cities from the videos below, or search for a specific city by the search bar, or go to the filter page by clicking on the CTA button. Now I am clicking on the big button on the left.



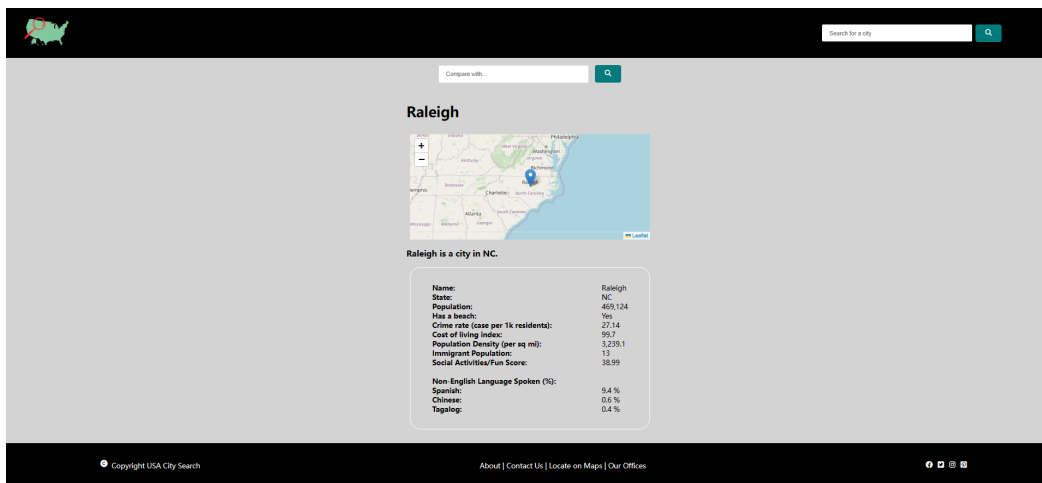
Here I am redirected to the filter & preferences page. On this page I will select some filters I care about and look for cities which meet this criteria. Since I am a student, I would like to move to a city with a low crime rate, low cost of living, and good schools. It should also be a fun city with lots of social activities to engage in. Therefore, I set the crime rate filter to “less than 30”, cost of living to “60-100”, and the top 75 colleges. To rank the cities which meet these criteria, I also set the preferences to “at least 40/100” on fun score and the state’s educational ranking should be in the top 5.



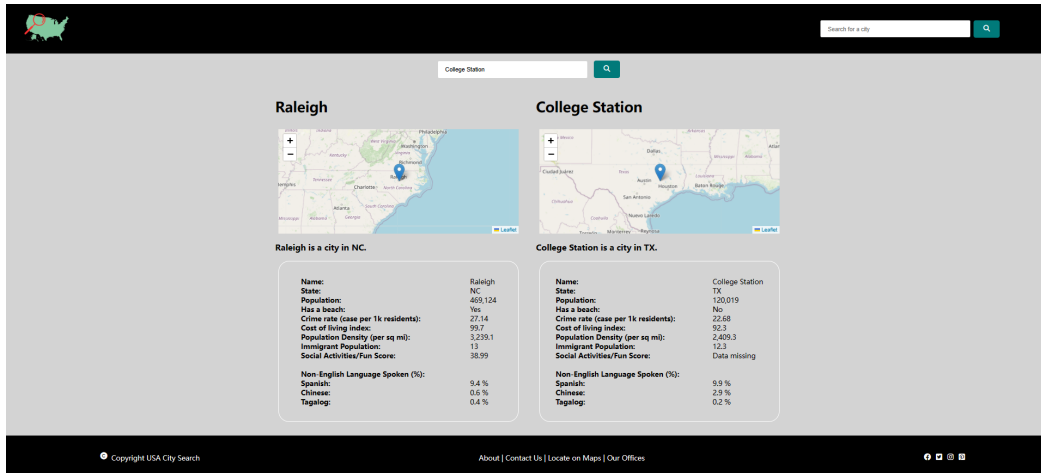
Here are the cities we got by the query. So we can see that College Station is the No. 1 choice for me from what I selected. But I changed my mind and I think the population in that city matters too. I want to live in a city with a larger population. So, I set the filters on the city list page, to filter the cities by population, and sort by descending order. Then I clicked the “search with added filters” button.



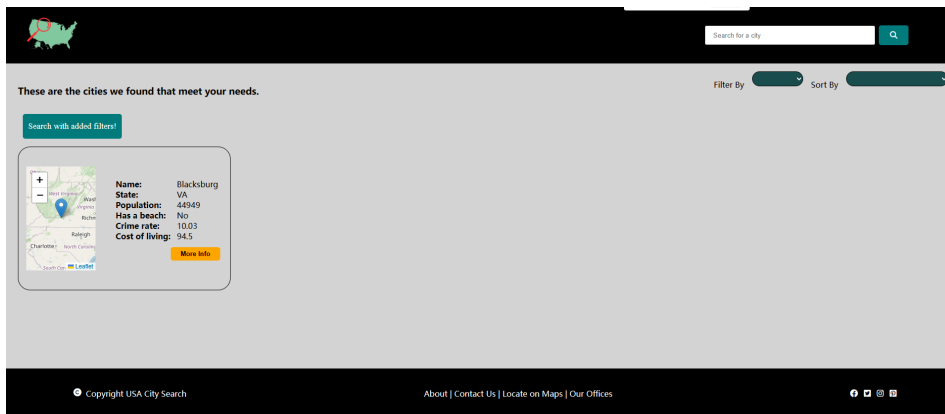
Now Raleigh became my first choice, and college station became the fourth since it has a smaller population. Now I would like to take a deeper look into Raleigh. So I clicked on the More Info button.



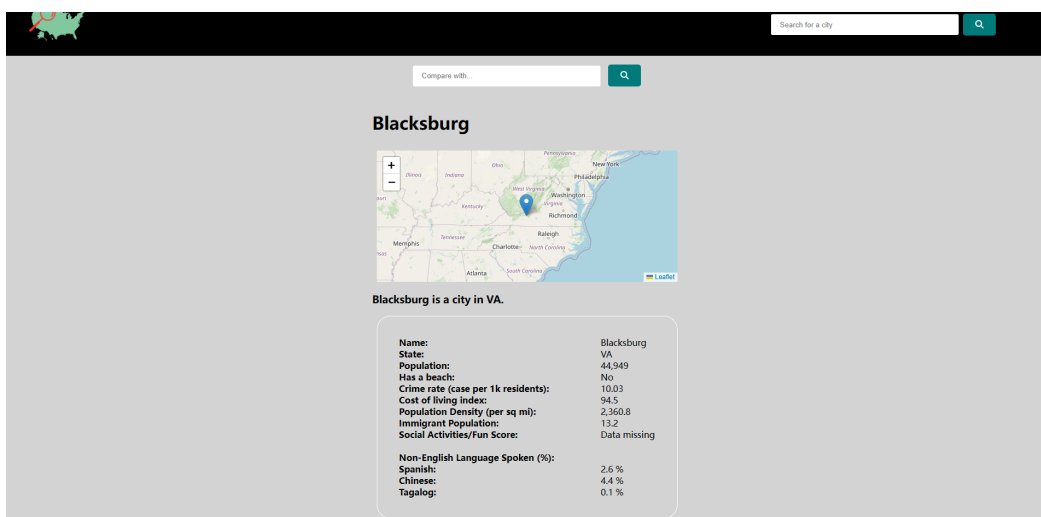
Here we can see more information about the city of Raleigh. But I still want to compare it with College Station, so I searched College Station using the “Compare with...” search bar.

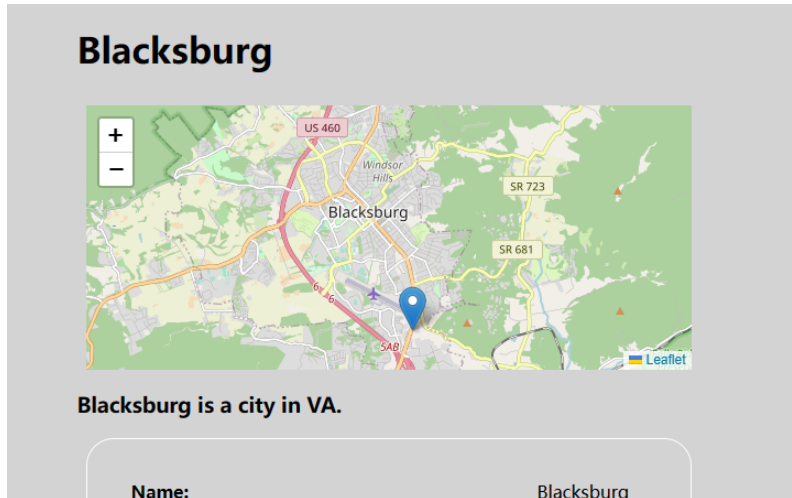


Here we can compare the two cities. But my dream school is Virginia Tech, so I want to check Blacksburg too. Now I entered Blacksburg in the search bar in the header, and hit the enter button on my keyboard. Here we go.



I clicked on the “More Info” button again and now we are here on the city information page.





I zoomed in on the map, now I can see what the town is like.

## 2.2 Features List

Our product provides the following features:

- The ability to search for cities using one or more filters and/or preferences to determine what cities appear in the results and how the results are ordered respectively
- The ability to search for a city by name and view its attributes
- The ability to directly compare two cities, viewing their attributes side-by-side
- The ability to view key attributes of each city that appears in the search results such as name, state, crime rate, population, and immigrant population
- The ability to view the locations of cities and nearby areas on an interactive map
- The ability to sort city search results by crime rate or population in ascending or descending order

## 2.3 User Stories

The following table shows all of the user stories we had planned for this project:

User Story	Tests (DoDs)	Priority	Tasks (with time estimate)*
As a user, I want to see the home page load as soon as I open the website so that I am able to view which filters to choose so that I can decide the place to live in next.	1. Navigate to the website, verify that city filters are visible	1	1. Download necessary packages (3 hr) 2. Create a "hello world" website (2 hr) 3. Add basic city database with cost of living information (5 hr) 4. Implement ability to filter city data based on conditions (5 hr) 5. Implement the cost of living filter (1 hr)

<p>As a software engineer who doesn't like taking risks, I want to be able to click on filters such as Crime Rate and an occupation such as CS so that I can decide if that place is suitable for me to live in.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by crime rate &lt; 10/100,000, check that all results meet this condition</li> <li>2. Filter cities by occupation = CS, check that all cities in the results have CS jobs</li> <li>3. Filter cities by both crime rate &lt; 10/100,000 and occupation = CS, check that all results meet both conditions</li> </ol>	5	<ol style="list-style-type: none"> <li>1. Add occupation to the city database (2 hr)</li> <li>2. Add occupation as a supported filter (1 hr)</li> </ol>
<p>As a libertarian I want to be able to click on filters such as the Income Tax rate so I can avoid giving too much of my hard-earned money to the government.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by highest income tax rate &lt; 30%, check that all cities in the results are from states meeting this condition</li> </ol>	15	<ol style="list-style-type: none"> <li>1. Add income tax field to the states table (1 hr)</li> <li>2. Add income tax rate to the list of supported filters (1 hr)</li> </ol>
<p>As a gun owner who recently got a remote job, I want to use City Search to find a city with less restrictive gun laws so I won't face burdens I consider excessive in owning a gun.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by concealed_carry = "shall issue", make sure all cities in the results have shall issue concealed carry laws</li> </ol>	14	<ol style="list-style-type: none"> <li>1. Add a states table with a concealed carry laws field (3 hr)</li> <li>2. Add concealed carry laws to the list of supported filters (1 hr)</li> </ol>
<p>As a parent, I want to find a city that preferably has good schools so that my kids can be well-educated.</p>	<ol style="list-style-type: none"> <li>1. Set the preference value for "good schools" to 10, make sure the cities with the highest-rated schools appear first in the results</li> </ol>	6	<ol style="list-style-type: none"> <li>1. Implement ability to rank cities based on preferences (5 hr)</li> <li>2. Add "good schools" preference with corresponding importance setter (1 hr)</li> </ol>
<p>As an extroverted user, I want to find a city that preferably has a high population density and lots of things to do (lots of things to do being more important) so that I don't have to sit at home all the time and get bored.</p>	<ol style="list-style-type: none"> <li>1. Set "population_density &gt; 1000/mi<sup>2</sup>" as a preference with importance 1; set "social_activities = 'high'" as a preference with importance 2; make sure that the results show cities with lots of things to do and a high population density, especially the former, first</li> </ol>	11	<ol style="list-style-type: none"> <li>1. Add population density to the database (2 hr)</li> <li>2. Add social activities ranking to the database (2 hr)</li> <li>3. Add population density as one of the preferences (1 hr)</li> <li>4. Add social activities ranking as a preference (1 hr)</li> </ol>

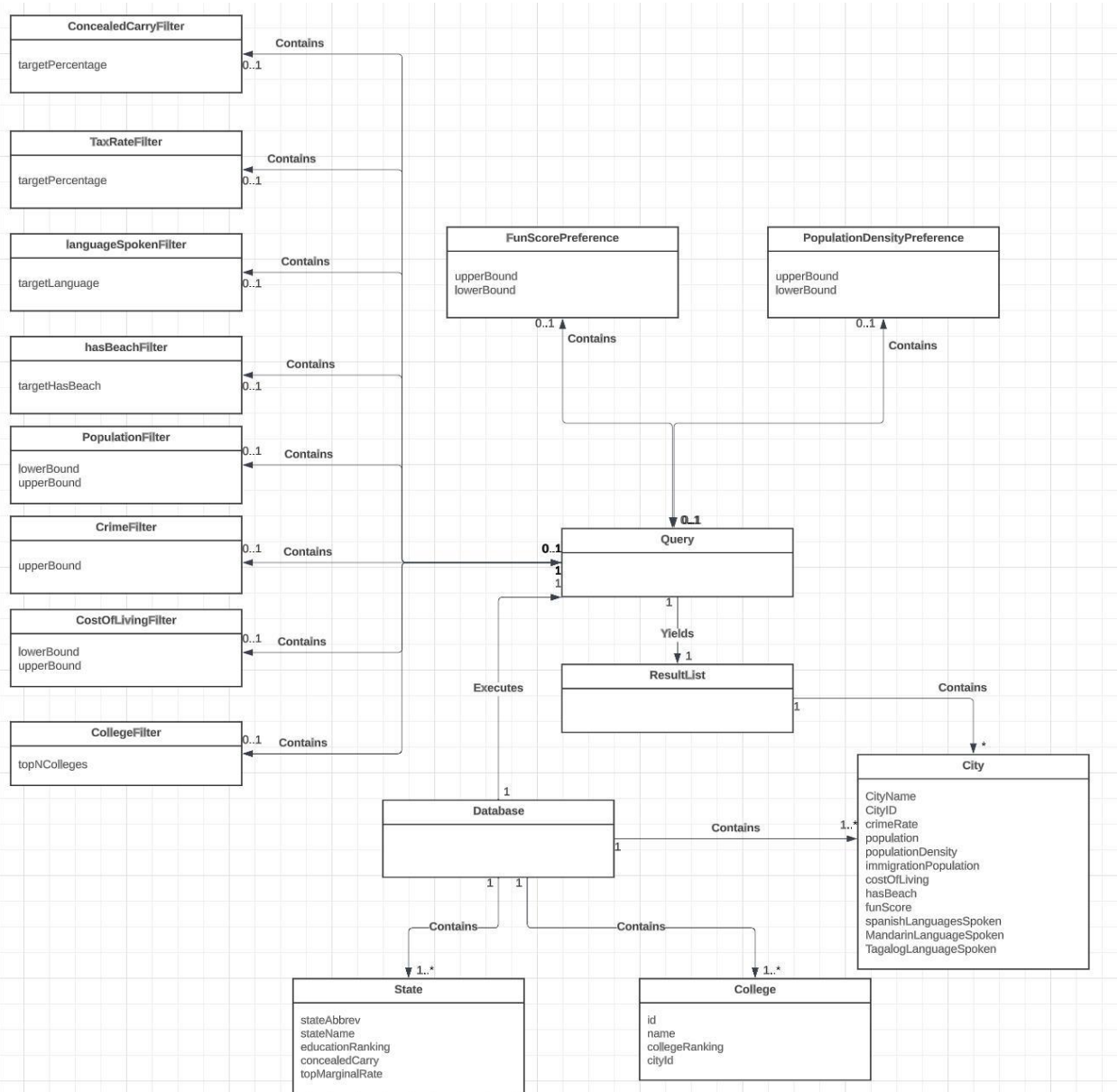
<p>As a housewife, I want to see the result with my selected filters in a clear and easy-to-read way so that I can spend less time getting the information I want and compare two cities easily.</p>	<ol style="list-style-type: none"> <li>1. Set one of the city filters; ensure that a third party who did not work on this feature can effectively understand and interpret the results</li> <li>2. Use the site to compare two different cities; ensure that statistics about each city are clearly visible to allow for easy comparison</li> </ol>	<p>10</p>	<ol style="list-style-type: none"> <li>1. Implement a city comparison tool (4 hr)</li> </ol>
<p>As a hispanic American, I want to be able to find a city with a large Spanish speaking population so that I can connect to my community.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by spanish_speakers &gt; 100,000; ensure that only cities with at least this many Spanish speakers appear in the results</li> </ol>	<p>12</p>	<ol style="list-style-type: none"> <li>1. Add language speaker population information to cities table (2 hr)</li> <li>2. Add language speaker population filters (2 hr)</li> </ol>
<p>As a mother with three kids, I want to be able to see all the public schools and the rankings in that city so that I can choose where I can send my kids to and where I should live.</p>	<ol style="list-style-type: none"> <li>1. Search for a specific city; make sure the user can view all public schools in that city along with their rankings</li> </ol>	<p>7</p>	<ol style="list-style-type: none"> <li>1. Add public school table (3 hr)</li> <li>2. Add public school rankings as one of the filters. (1 hr)</li> </ol>
<p>As a high school student, I want to clearly check the colleges and their rankings in each city, so that I can choose my preferred schools.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by college_ranking &lt; 100; ensure only cities with colleges ranked in the top 100 are shown in the results</li> </ol>	<p>8</p>	<ol style="list-style-type: none"> <li>1. Add college table (3 hr)</li> <li>2. Add college ranking as one of the filters (1 hr)</li> </ol>
<p>As a student who expects to develop in the field of chemistry in the future, I hope that there are companies or institutions in related industries near my school. I want to use a similar occupation filter to get relevant information.</p>	<ol style="list-style-type: none"> <li>1. Filter cities by has_chemistry_degree=true; ensure that only cities with colleges offering chemistry degrees appear in the results</li> <li>2. Filter cities by chemistry_job_openings &gt; 100; ensure that only cities meeting this condition appear in the results</li> </ol>	<p>9</p>	<ol style="list-style-type: none"> <li>1. Add degrees offered information to the college table (2 hr)</li> <li>2. Add has_degree filters (2 hr)</li> <li>3. Add job_openings information to the cities table (2 hr)</li> <li>4. Add job_openings filters (2 hr)</li> </ol>
<p>As a young woman with little income, I hope to use the crime rate and cost of living filters to find a low crime rate and low</p>	<ol style="list-style-type: none"> <li>1. Filter cities by crime_rate &lt; 10/100,000; ensure that only cities meeting this</li> </ol>	<p>2</p>	<ol style="list-style-type: none"> <li>1. Add the crime rate of each city to the database (2 hr)</li> <li>2. Add a crime rate filter (1 hr)</li> </ol>

cost of living area to live in.	<p>condition are present in the search results</p> <ol style="list-style-type: none"> <li>Filter cities by <code>cost_of_living &lt; \$5,000/month</code>; ensure that only cities meeting this condition are present in the search results</li> <li>Filter cities by both <code>crime_rate &lt; 10/100,000</code> and <code>cost_of_living &lt; \$5,000/month</code>; ensure that only cities meeting both conditions are present in the search results</li> </ol>		
As a retired woman, I want to find a city which is near the beach, less populated and crime free so that I can relax in a calm and peaceful place for my remaining years.	<ol style="list-style-type: none"> <li>Filter cities by <code>has_beach=true</code>, <code>population &lt; 100,000</code>, and <code>crime_rate &lt; 10/100,000</code>; ensure only cities meeting all conditions appear in the results</li> </ol>	3	<ol style="list-style-type: none"> <li>Add <code>has_beach</code> field to the database (2 hr)</li> <li>Add population of each city to the database (2 hr)</li> <li>Add <code>has_beach</code> filter (1 hr)</li> <li>Add population filter (1 hr)</li> </ol>
As a grandparent, I want a small town with a low cost of living so that I have a large house to accommodate my whole family when they come visit me.	<ol style="list-style-type: none"> <li>Filter cities by <code>cost_of_living &lt; \$5,000/month</code> and <code>population &lt; 100,000</code>; ensure only cities meeting all conditions appear in the results</li> </ol>	4	None (tasks needed for previous user stories are sufficient to satisfy this story)
As an immigrant, I want to find a place to live that preferably has other immigrants so that I can find people who share my culture.	<ol style="list-style-type: none"> <li>Filter cities by <code>"immigrant_population &gt; 20%"</code>; ensure only countries with at least 20% immigrants appear in the search results</li> </ol>	13	<ol style="list-style-type: none"> <li>Add <code>immigrant_population</code> to the database (2 hr)</li> <li>Add <code>immigrant_population</code> to the list of filters (1 hr)</li> </ol>

We managed to implement all of the user stories by the end of the semester except for those with priority 5, 7, and 9. This is because we didn't manage to find data showing the amount of job openings in each occupation in each city, the degrees offered by each college in our colleges list, or a list of all the public schools in each city. We did, however, find the data necessary to complete all the other user stories. Links to the sources where we got all our data can be found in the comments at the top of the `data.sql` file under `server/src/main/db` in the project code.

### 3. Design

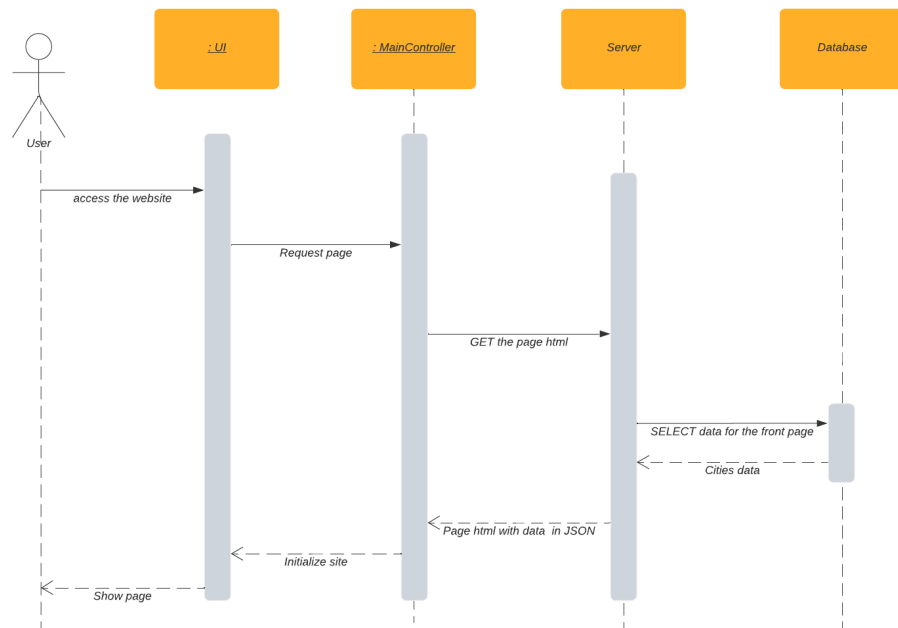
#### 3.1 Domain Model



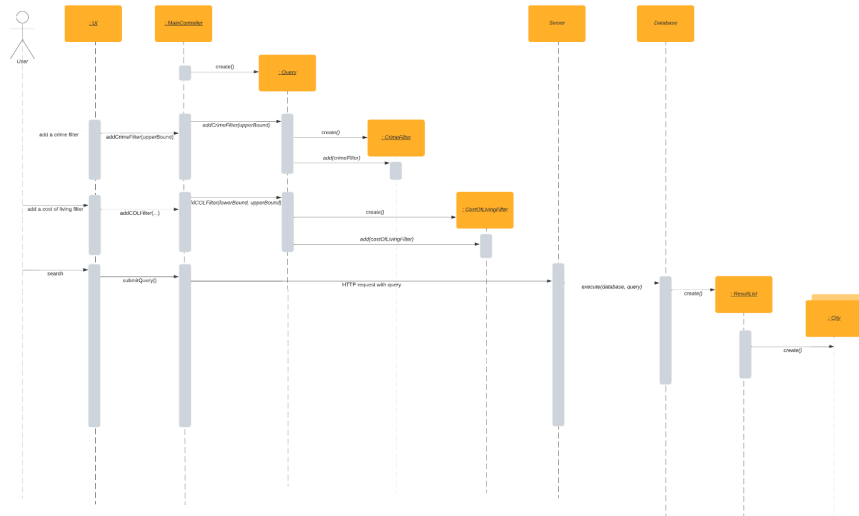
The domain model provides a brief overview of how the project functions; on the left, it lists a number of filters the user can choose from, each of which can be included in or excluded from the query. This query is executed on a database that contains the three tables City, State, and College. State names, state abbreviations, education rankings, concealed carry laws, and top marginal rates are among the columns in the state table. The college table includes columns for the city ID from the city table as well as columns for the ID, college name, and college ranking. CityName, CityId, Crime Rate, Population,

Population Density, Immigration Population, Beach, Fun Score, and Languages Spoken, including Tagalog, Mandarin, and Spanish, are some of the columns in the city table. In addition, the domain model includes preferences for the user to choose from, such as Fun Score Preference and Population Density Preference. Executing the query on the database yields a ResultSet that contains a list of the cities matching the query parameters.

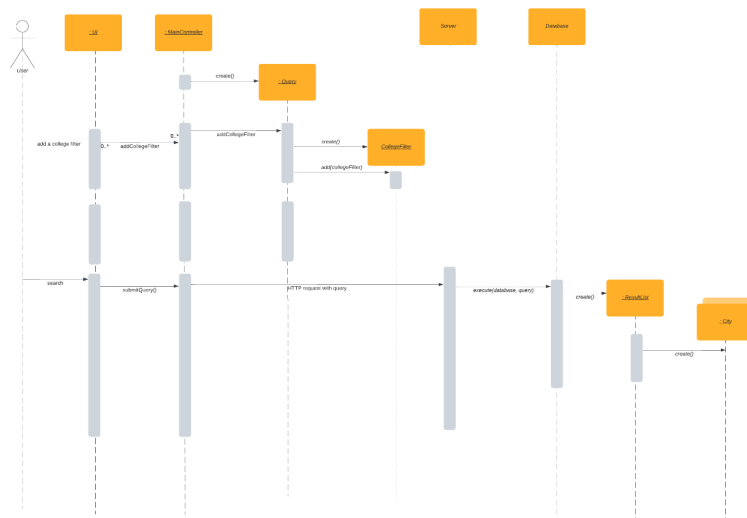
### 3.2 Sequence Diagram (View filter page on website)



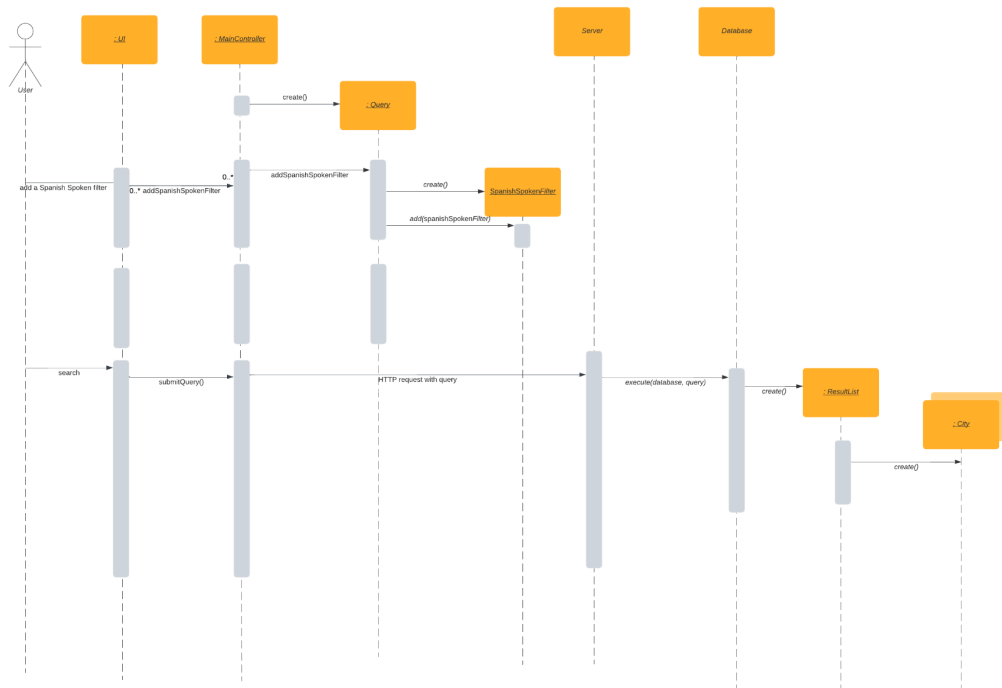
### 3.3 Sequence Diagram (Simultaneous crime filter and cost of living filter)



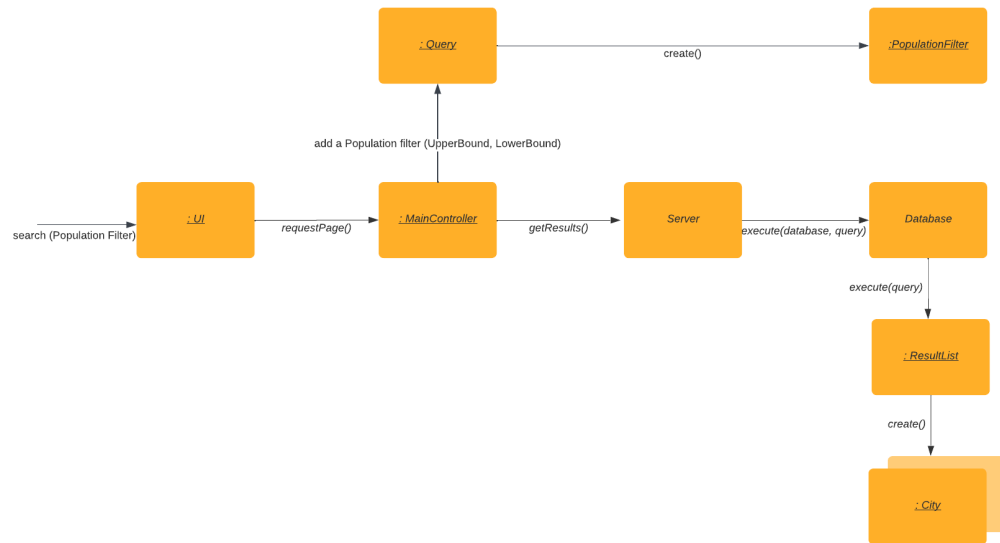
### 3.4 Sequence Diagram (College filter)



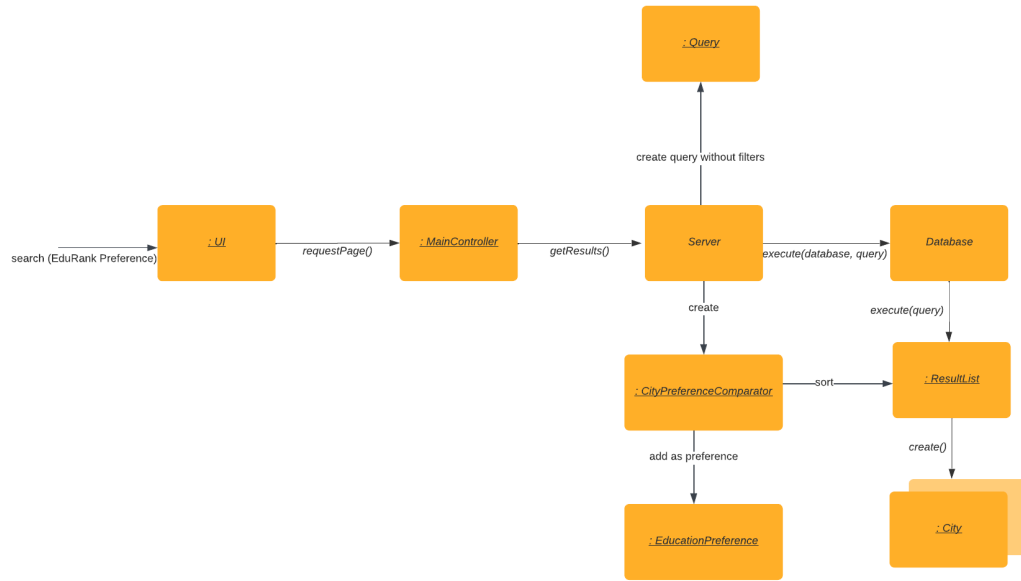
### 3.5 Sequence Diagram (Spanish spoken filter)



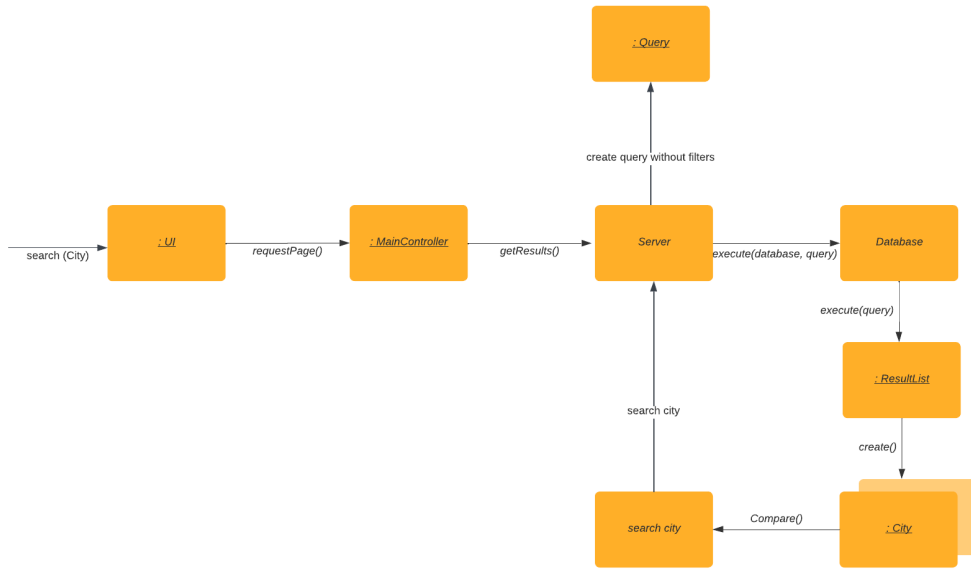
**3.6 Collaboration Diagram (Population filter)**



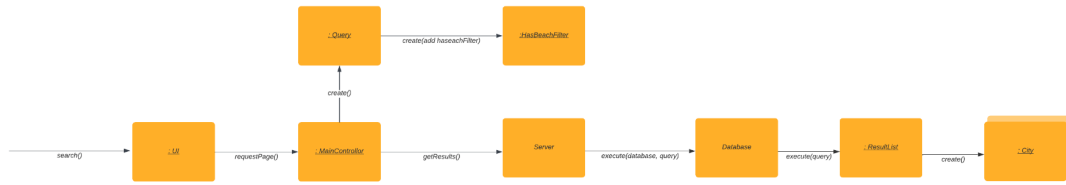
### 3.7 Collaboration Diagram (Education ranking preference)



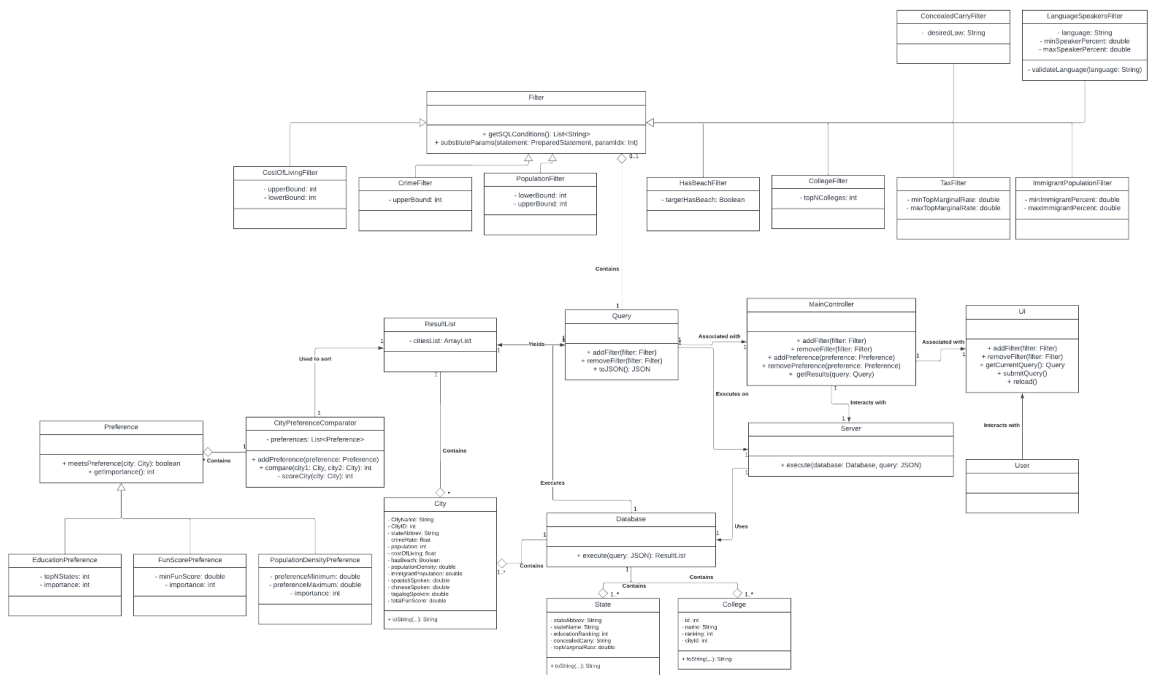
### 3.8 Collaboration Diagram (City comparison)



### 3.9 Collaboration diagram (Has beach filter)



### 3.10 UML Class Diagram



### 3.11 Frontend

Before starting to code the frontend, we created a Figma design in order to have a blueprint on how the website should look like. With the aid of Typescript and CSS, we

implemented the front end using Vue. We specifically created AppHeaderVue and AppFooterVue (representing the website header and footer respectively) as separate components so that we could import these in other pages rather than having to create the app header and footer in each page. Several components are used by the JavaScript framework Vue to create Single-Page Applications (SPAs). We also made use of different input components for the user to have a combination of filters to make the user more interactive with the website. We made use of CTA (call to action) buttons as one of the most important features or components to have on the homepage of the website. Apart from that, we added a few videos that matched the theme of the website that displayed 30 seconds to a minute video of different cities.

We used Pinia stores to manage the global state of the website. When the query is concatenated and sent to the backend, a list of cities would be returned back to the frontend and saved in our city store. The city list page only contains the panel of the cities. On the panel, we loop through the cities and put each of them into a city card component to show some brief information about each city. Since a city card has limited space, if the user wants to look up all information about a specific city, can look at the city information page. On this page, we also have a comparison search bar so users can compare the currently selected city to another one. The user posts the name to the backend, and the city to be compared would be returned and stored in the comparison store. The key concept of the city list page and city information/comparison page design is to maintain high reusability by creating and reusing components on each page. That improves the readability and maintainability of the code.

### **3.12 Backend**

The backend is the portion of the application that is responsible for receiving filter and preference information from the frontend and returning a list of matching cities to the frontend. The first step in completing this task is to generate a SQL query from the filter information. To help accomplish this, we created a Filter interface with `getSQLConditions()` and `substituteParams()` methods. The `getSQLConditions()` method is used to get a list of conditions in SQL syntax (e.g. “population >= ?”) that correspond to the filter conditions, and then all the conditions from all filters are combined into one SQL query. Then, the `substituteParams()` method of each filter is called to substitute the user data into the query. Each filter subtype is coded to implement `getSQLConditions()` and `substituteParams()` differently. After creating the query, the query is executed on the database to get a list of city results.

We chose this multi-step process for creating the query to avoid concatenating raw user data directly into the query, which could have made our website susceptible to SQL injection attacks.

The second step that the backend performs is ordering the results based on user-specified city preferences. It does this using a Java Comparator object which is used along with the Java sort() function to sort the cities. This comparator compares two cities by “scoring” them based on which city best meets the user’s preferences. To come up with the score for each city, the comparator loops through a list of Preference objects representing the user’s preferences, uses the meetsPreference() method on each one to check if the city meets the given preference, and increases the city’s score by the preference’s importance value if the preference is met. The end result is a filtered and ordered list of cities that can be returned to the frontend.

### **3.13 Tools**

We developed the frontend for our website using the Vue framework, and we developed the backend using Tomcat and Jakarta EE. We also used MySQL for the database. These tools were used because many members of our team already had experience with these tools and frameworks from the web development class, allowing us to focus more on developing the product and less on learning new tools.

### **3.14 Unique Features**

There are other products available that people can use to help find a new place to live such as BestPlace, Livable, and Teleport. The main thing that makes our product unique is its ability to combine multiple filters and preferences together in a single, complex query, helping users find a city that meets their exact needs. Many of the existing products such as The Earth Awaits rely exclusively on filters, and other products such as Teleport use preferences but require the user to create an account to customize their preferences.

## **4. Retrospection**

### **4.1 What went well and what went wrong**

During the project, we did a good job of holding meetings and communicating regularly, which made it easy for people to follow what was going on. We also used Trello to delegate tasks, especially during sprint 3, which helped everyone be productive. In addition, we came up with a design for the backend that makes it really easy to add support for new filters and preferences to the REST API, which was very helpful in the later sprints.

While working on the frontend there was some confusion over the difference between filters and preferences and the best way to communicate these differences to users through the user interface. Fortunately, however, we decided on a design clearly

differentiating these features by the end of the semester. Another difficulty is the fact that a few of our user stories required data that was too hard to access or too volatile to practically use in the project, which required us to skip these stories.

## **4.2 What we learned**

Shreya - I learned more libraries in Vue for frontend development and learned different ways to receive input from users.

Kenneth - I learned how to debug issues with the IntelliJ project setup and Tomcat run configurations. This project also gave me more experience with SQL and the Java SQL library, and I learned how to communicate with members of a technical team to complete a software project.

Yingfeng - In this project, I learned a complete program development process based on scrum. I determined the priority of user stories through the product backlog, and then formulated the program functions that need to be implemented at each stage in the roadmap according to the priority.

Pallavi: I learned how to work with both frontend and backend and connect client and server. In this project I learnt how to collaborate with team members and work efficiently in a team. I also learned how to work with a version control tool.

Joseph - I learned how to fetch data from the backend to the frontend and listen to the async function dynamically. I also learnt how to work in a self-managed scrum team. I also had a better understanding of Typescript and Vue.js features.

## **4.3 Actions that should have been taken**

I think we could have avoided some problems by thinking more deeply about our proposed user stories before including them in our list. Specifically, we should have been more careful to make sure that the data required for all our user stories was feasible to access and unlikely to change frequently. We should also have started using Trello earlier in the project so that everyone knows what each team member is working on at all times.

## **4.4 Improvements**

We could improve our project by adding more cities and colleges to the database. An automated script in particular could vastly increase the size of the database without much additional human effort, allowing search results to be more complete. We could also design the frontend to be more flexible. For example, having sliders that allow the user to independently select minimum and maximum values for the numerical filters could allow for filter settings not currently supported by our dropdowns and radio buttons.

Additionally, we could add autocomplete to the parts of our website that let users search for cities by name.

## **5. GitLab**

Link - <https://code.vt.edu/pallavi21/city-search>