City of Windsor Open Data Group Project:
Final Technical Report

Submitted By (Group 12):

Saffa Alvi
alvi118@uwindsor.ca

Alina Noor
noor112@uwindsor.ca

Lama Khalil
khali121@uwindsor.ca

COMP-3220: Object-Oriented Software Analysis and Design
Dr. Ziad Kobti
August 12, 2020

2. Abstract

The objective of this City of Windsor Open Data project is to improve the existing open data portal to better facilitate data collection, management, distribution and documentation with high quality application programming interfaces (APIs). These APIs will allow programmatic access of the City of Windsor open datasets so they can be used to create valuable applications. This project was focused on accessing and manipulating two datasets from the City of Windsor Open Data Catalogue, Community Centres and Property Flooding/Grading Service Requests, as the foundation to create practical programs for the rest of the open datasets. These datasets were loaded onto the program, displayed to the user, then extra functionalities such as sorting, calculating distance and filtering service request history were added to interact with the dataset information. This program was enhanced through the project phases as it was first tailored specifically to each of these two datasets and later refactored to be compatible with different file formats. The resulting product of this project was an application that could be used to load any dataset from the City of Windsor Open Data Catalogue which could then be altered to add data-specific features in the future.

3/4. Keywords

Open data, dataset, community center, property flooding, service requests, API, community centres, distance, near, flood zone inquiry, finder, street flood history.

5. Acknowledgment

We acknowledge Dr. Ziad Kobti for providing the class with an example FileManager application that was referred to for the creation of our third phase working product. The example application helped in creating the foundation of our second elaboration iteration implementation. We would like to recognize the City of Windsor Open Data Catalogue for providing the two open datasets, Community Centres and Property Flooding/Grading Service Requests that were used to create and test the programs. We would also like to recognize GeoDataSource as their method of calculating distance between latitude and longitude was referred to in order to create the Distance class in the Community Centres dataset program [1].

We would like to thank all the group members for completing their respective tasks efficiently and collaboratively. For Phase One of this project, the inception artifacts were evenly split and later reviewed, as a whole, by all the team members. For Phase Two and Three, Saffa and Lama handled all tasks related to the two datasets, Community Centres and Property Flooding/Grading Services respectively, such as creating the working code implementations, designing UML diagrams, writing descriptions and setting up the GitHub repositories. Alina completed all tasks related to the testing of the working code implementations in BlueJ as well as setting up and updating the documentation repository on the MyWeb Server. The reports for these two phases were also evenly distributed between the members and reviewed, as a whole, upon completion. The PowerPoint presentation for this final phase was created and compiled by Saffa and all team members contributed by recording voice narrations for the slides. Each member updated the

project board with detailed descriptions of the breakdown of tasks which can be found on the project management board on GitHub.

6. Introduction

The City of Windsor Open Data Catalogue was created with the intention of data reuse for research purposes and to encourage citizens to improve their interaction with municipal services and facilities [2]. The objective of this project is to enhance this existing website by making it more user-friendly and interactive so users can easily access and visualize public data in several formats and use the dataset APIs to create useful applications. The motivation behind this project was to allow the City of Windsor residents to have access to public information about the city and its infrastructure, to encourage research and innovation, help them make informed decisions, and learn more about the city. This project will also encourage developers in the city to access these datasets and public resources and create some alternate use for them to benefit the citizens of Windsor when they need relevant and important information.

The idea of open data is to make data related to individuals, establishments and/or governments available for general use to the public. This data can contain anything about the area and citizens such as the locations of local businesses, curb maintenance service requests, election results, etc. It would be easily accessible, free, and easy to reuse and repurpose. Gathering all this data and making it available to the public in one place will make it easier to access and encourage citizens to seek more information about topics that relate to their everyday lives. It also creates a gateway for developers to use this opportunity to develop useful applications with these convenient resources to benefit the citizens of their community. We hypothesize that a user-friendly open data portal that has useful applications with the datasets will motivate the residents to learn more about the region in which they live and gain more knowledge about topics they encounter in their daily lives. It will also support developers who want to help users analyze these datasets quicker and more effectively.

This report follows our project progress and is broken down by three phases. The first phase consisted of an Inception Report to outline all the foundational information for beginning this project. It included the nine inception artifacts: vision and business case, use-case model, supplementary specification, glossary, risk list and management plan, iteration plan, phase and software development plan, development case, and prototypes/proof of concepts. This phase consisted of establishing a common goal and overview for the project, highlighting objectives and expectations, planning for next iterations, and preparing for the working implementation in the next phases.

Then we moved on to Phase 2, which was the first development cycle, Elaboration Iteration 1. This phase consisted of following the plan and objectives set out in the previous phase to create a working prototype. We created a proof-of-concept for one of the use-cases stated in our Inception Report which was “Accessing the API (For Developers)” to give the client a better understanding of what the basic product would look like. This phase was important in developing the core architecture of the product, determining the resources and feasibility of the project and representing the overall idea into a physical model for the client. This phase helped us recognize and resolve the high-risk elements, as well as, gave us a better understanding of

what the next phases of the project would look like and helped determine if our objectives set out in the first phase were truly achievable.

The third and final phase of this project was moving onto Elaboration Iteration 2. In this phase, we focused on enhancing our previous prototype to meet more specific requirements that were defined in the previous phase. We now wanted to use the base prototype to add in features that would take care of the main demands and high-risk elements that we couldn't complete in the previous iteration. We also wanted to set up this prototype to be easily adaptable and changeable for the next phases. This phase was focused on refactoring the code with GRASP patterns to ensure the most efficient design that would meet the client's requirements and expectations.

This report summarizes all the objectives, progress, key findings and results that were completed and discovered at the end of this project. The report is organized as follows. The next section talks about all related work that was used to prepare for the project. The eighth section describes our approach, methodology, and steps taken to complete this project. In the ninth section, we explained how we executed and what experiments we did to test our prototypes. The tenth section is an overall discussion of the findings and our thoughts on our experiments and prototypes. In the eleventh section, we present a conclusion which reflects on our objectives and what we did and did not achieve in this project. The twelfth section describes any future work we plan to develop and how we think it will benefit the market. Our last four sections, thirteen to sixteen, list the references used in this report, an appendix for the group work and project management, another appendix that contains figures/screenshots that are referenced in sections of the report and lastly, an acknowledgment section for platforms used.

7. Related Work

The research and understanding of the idea of open data was first explored including what it is and its purpose. We then moved onto examining a list of case studies mentioned in the project requirements to develop a set of categories, ideas, steps, recommendations for future work, and definitions. An effective open dataset API breaks down raw data and creates new levels of collaboration to derive information from datasets in a productive and reliable way. Such APIs serve as the single customized source for users to access the needed data more quickly.

The team proceeded to examine the first case study which was that of the current City of Windsor Open Data Portal. This is where the City of Windsor's vision and mission was analyzed. Since this project's focus was the City of Windsor's Open Data Portal, the team made sure to consider the current vision as part of the proposed vision for this project.

The team then moved onto the second study case which was that of the City of Toronto [3]. The City of Toronto has a very informative open data website and a detailed improvement and management strategy plan. It also has an "API for Developers" section on each dataset with code snippets available in different programming languages. For this project, the team decided to adopt this feature from this case study which can be seen in the project's prototype. However, their data catalogue was a little hard to navigate.

The key points taken away from the third case study, Canadian Health Institute for Health Information (CIHI), was that it was very concise and covered all major topics like vision, audience, goals and strategies [4]. These topics also inspired the team to think in a broader perspective for the phase and software development plan in terms of the audience, goals and strategies to use for this project.

Moving forward, the Government of Canada's [5] website for open data was explored and analyzed as part of the research conducted for this project. The platform provided immense detail and most importantly included a glossary and API library. The glossary was used as inspiration for the new City of Windsor's open data portal. It contained a lot of resources for beginners and described the required licencing and open datasets related legal matters.

The conclusion to our research phase included analyzing case study 5 which was that of the CKAN platform [6]. This case study provided a lot of information for both the platform's technical side and user-focused side. For the technical side, the team was inspired by the tools used to build the CKAN API. The highlight for CKAN was their CKAN-specific coding standards. On the user-focused side, the aim of the CKAN platform was very important to the project since the main goal was to provide a single point of access for open data. The CKAN platform also provided details on the potential challenges of creating a data portal and those challenges were very useful for the team to understand and find solutions to.

8. Approach

The following illustrates the approach flow for the project. This project began with an initial review of the current City of Windsor Open Data Catalogue, hereinafter referred to as "the website". It appears that the website only contains raw datasets. These datasets were collected directly from the source and have not been subject to processing. Presently, when the users need this information, they can download the datasets individually in separate files and different formats. While using the current self-serve system might be satisfying, it can often be time consuming. The users can not customize their search to get the desired data in a short time. This is where an effective open dataset API solution becomes essential.

After reviewing all the case studies mentioned in the previous section, the next step was to explore the Artifacts of Inception. This step was an absolute necessity before solidifying the main idea for our proposed prototype. Using the Inception Report, we were able to recognize the risk factors and develop a management plan. We were also able to define the project glossary, figure out the process of obtaining the required legal licences, and set up the technical requirements along with creating use-case models, listing supplementary specifications, and scheming the iteration and phase plans.

Based on our observations from the above-mentioned five case studies, our inception report, and keeping inline with the City of Windsor's vision, the algorithm leading to our prototype was focused on facilitating access to the datasets, developing useful applications to use the datasets such as increasing searchability within them, and providing options for data visualization.

Since all the team members were most familiar with the Java programming language, we decided to use this language to create our working code prototype. We thought that using this object-oriented language would be the best way to effectively produce our vision for this prototype and the use of classes would facilitate more focus and better understanding of each component of the software. IntelliJ was an IDE that we all had the most experience using when programming with Java and it included built-in developer tools such as version control, testing, debugging and terminal access. For these reasons, we decided this would be the best environment to use to keep the programs consistent across various platforms and allow all editing and testing aspects of the code to be done in one place.

Our approach for the second phase (elaboration iteration 1) was to complete an application that would meet the most basic requirements for our vision which was to load a specific dataset, display the information and make some practical use of the data. We were not focused on creating a program to fit every single dataset on the website but rather, focused on gaining that foundational understanding of how the program interacts with the dataset to extract information and how we could utilize it to fulfill our own purpose.

The algorithm we developed for this phase was using a *Main* class to load the dataset, display the menu, and process user responses to the program. We then used a *Centres* or *Floods* class to contain all the fields and attributes of that specific dataset. Lastly, to add in an extra functionality to calculate distances (for Community Centres) or find a specific street in the service requests (Property Flooding/Grading Service Requests), we added in an extra class, *Distance* or *Street* respectively. This program was heavily structured for these specific datasets and to use any other dataset file, the majority of this program would have to be altered, specifically the *Centres* and *Floods* classes. Since we were just trying to gain a better understanding of loading and processing datasets, this algorithm served our purposes and completed our vision for this phase. Once we had gained the necessary knowledge from experimenting with these two specific datasets, we were ready to move onto the next phase.

Going into phase three, we now had a better understanding of the background processes that were needed to create powerful APIs for the datasets due to the experiments done in the previous phase. Our main objective of this phase was to now use the knowledge acquired from the previous phase to alter our program to be compatible with any dataset from the website. We referred back to the FileManager application example that was provided by our instructor, Dr. Ziad Kobti, in class, to create the base for our program. Once we were able to use that same program with changes to only one interface class to work for dynamic datasets, we then went in and added dataset-specific features and applications to enhance the functionality of our program.

The algorithm we developed for phase three had many differences from the previous iteration. We refactored our program so that if the user wanted to load a different dataset, they would have to make the most minimum changes possible since dependency between the classes would now be very limited. We now have classes that had specific tasks assigned to them which would make it easier to navigate between the program and understand the software components. We used a *Template* interface to store all necessary information about the dataset. In order to access different datasets, this class would be the only component that would need to be altered to fit the

data. Our *Load* class was responsible for loading all the information from the file while the *Record* class was used to create and store the individual records loaded from the file. We had the abstract *Visualizer* class that was used to create the list of records loaded from the file which are then ready to be displayed to the user. This class would also be used if further visualization options were given to the user, however we only implemented the console view in this iteration. The *UserInterface* class was used to display outputs and handle all interactions with the user while the *Main* class was used to create the instance of the *UserInterface* and start the program. This algorithm was designed to work dynamically with any dataset that the user would like to access.

9. Experimental Setup or Demonstration

For *Phase 1*, completing the inception report provided the team with a better overview of the requirements. This was a powerful tool for creating the project's prototypes. The prototypes included facilitating access to the datasets and developing useful applications to use the datasets such as increasing searchability within them and providing options for data visualization.

The inception report was useful to discover potential issues and risks at an early stage of the project. However, our initial estimation of what we could accomplish was overly ambitious. After we became more involved in developing the prototypes, we realized that our team did not possess the required knowledge and experience to meet our hoped-for aspirations by Phase 3.

Therefore, our plan to show different visualizations of the datasets such as through graphs or on Google Maps, was not completed by Phase 3. We also planned on implementing a "related dataset" feature page but were unable to complete this due to time constraints. These are goals we hope to achieve in future iterations. Figures 1-3 in the Appendix provide more details on the result of this phase.

Based on the first iteration requirements, the development team worked to deliver a working software within the prescribed deadline of July 16, 2020.

The team adhered to the following workflow to complete *Phase 2* (Elaboration Iteration 1):

1. Defining the tasks required for the iteration based on the iteration plan.
2. Scheduling recurring team meetings to maintain open communication, distribute tasks, update on progress, and keep the iteration moving forward to meet the deadline.
3. Keeping in touch with the stakeholders (Dr. Kobti in this case) to involve them in the process and receive their feedback.
4. Setting up the team collaboration tools, such as GitHub for code repository, version control and project management/task tracking for the team. GitHub and IntelliJ were set up for bug reporting and the MyWeb Server was used as the documentation repository and the Software Architecture Document.
5. Drawing the architectural designs for the use-case model, "Accessing the API for Developers", using UML diagrams such as: Domain Model, Design Model (Class and Sequence Diagrams), Data Model and Use Case Diagram.
6. Creating a list of the required algorithms based on the UML diagrams.

7. Deciding which algorithms we have enough knowledge to code and which ones we needed to learn or conduct some research on.
8. Creating test cases for our two APIs for the two datasets from the City of Windsor Open Data website.
9. Designing the most important algorithms first.
10. Using the Java programming language to code both APIs based on UML diagrams, test cases and algorithm designs.
11. Creating code descriptions in the documentation repository.
12. Testing the documentation repository by asking outside sources if it is easily understandable and navigable to them.
13. Testing and debugging the code implementations using BlueJ. It is worth noting that, in both elaboration iterations, testing the APIs with all classes interacting with each other was successful in IntelliJ. However, when we tried unit testing with BlueJ the testing of some classes was not successful and we were not able to find a solution. Since the Distance and Findflood and Street classes used attributes from other classes as their parameters, we were unable to figure out how to unit test these classes since they could not access those external arguments.
14. Writing a summary of this iteration.
15. Updating the next iteration plan.
16. Delivering working softwares.

By the end of this iteration, we accomplished one of our prototypes discussed in the inception phase (API for Developers). We developed two working APIs for two datasets that were downloaded from the City of Windsor Open Data website.

The first API loaded data from the .CSV file for Community Centre locations from the local computer and allowed the user to use this data to find the nearest community centers to a location they entered. The main algorithm here was loading the data and computing the distance between the latitude and longitude points of all the community centres and a user entered location. The second API loaded data from the .CSV file for Property Flooding/Grading Service Requests from the local computer and allowed the user to use this data to find requests made about a street they entered. The core of this API was loading the data and implementing the search algorithm. More information about the implementation details can be found on our code repository on GitHub.

In this second phase, we had planned to be able to access the data straight from the City of Windsor website so it would be dynamic and the user would not have to download it to their local computer. We also hoped to be able to map the data on Google Maps and calculate distance (for community centre locations) based on the city map. However, due to time constraints, lack of applicable knowledge in these areas and some resources, we were unable to implement these into this iteration. Because we could not calculate the distance by considering the street map of Windsor, we opted to calculate the straight-line distance between the pair of latitude and longitude points to provide the user with a brief idea of the distance. We also decided to access and load the data from the local computer because we were unable to configure the program to access it straight from the website when testing both implementations.

The results we accomplished in this iteration were that we delivered a working code that displayed the required data based on the user's entry. The Property Flooding API can be run and executed through the terminal/console. After compiling all the files, the user needs to run the Main class. They will be prompted to enter the filename, which in this case will always be the PropertyFlooding.csv file straight from the City of Windsor open data portal downloaded to their local computer. It will then display the number of loaded records from the file and display a menu to the user from which they can choose options. If they choose 1, all loaded records will be displayed. If they choose 2, they will be prompted to enter a street name which they want to search the service requests for. The program will then display a list of service requests that were made about the same street entered. In the Appendix, Figures 4-5, we show the result of the user choosing to display the service requests with a specific street (Wyandotte in the example) as well as displaying all the loaded records.

Similarly, the resulting product of the Community Centres API can also be executed from the terminal. The process of compiling and running is the same as mentioned above in the Property Flooding API. However, in this API, if the user chooses 2, they will be prompted to enter the latitude and longitude of an address they want to find the nearest community centres from. The program will then display a list of community centres with a straight line distance of 5km or less from that address. In the Appendix Figure 6, we show a few examples, where the user first chooses to display all the loaded records and then chooses to display the list of community centres that are 5km or less from an entered address (University of Windsor in this example). The user then enters 0 to exit the program.

The goal of the second elaboration iteration, *Phase 3*, was to improve the prototype completed in the first iteration. In the first elaboration iteration, the code did not adapt any of the General Responsibility Assignment Software Patterns (GRASP), due to the team's lack of applicable knowledge in this area. However, this was implemented during the second elaboration phase. The following section will provide more details on the five GRASP patterns used to improve our APIs:

I. Low Coupling

Coupling is a measure of how strongly one element is connected to, has knowledge of or relies on other elements in the design. Low Coupling supports low dependency and low change impact between classes. In our first elaboration iteration, the methods and attributes in the Floods, Street, Centers and Distance classes helped lower dependency between classes, however it did not eliminate it. These classes maintain low coupling between the file structure and the searching functionalities to a certain degree. For example, when changes were made to the Community Centers API so it would be able to read the flood services CSV file, we had to make changes in all the classes. Such changes had a small impact on other classes in the system. However, we realized later that our code did not fully eliminate dependency. Hence, we refactored according to the Low Coupling design pattern. In our second elaboration iteration, we changed the code, such that changes to the attributes have to happen only in the Template class (a global interface) in order to use the code to read different files. The changes will have no impact on other classes in the system. Therefore, increasing our code's reusability and helping it better maintain Low Coupling.

II. Polymorphism

Polymorphism is defined as the condition of occurring in many different forms and this pattern handles how a general purpose is allocated to a set of classes or interfaces. Using this design pattern in the second elaboration iteration provided us with solutions to how we could create a pluggable software component and how to handle alternatives based on types. For example, Visualizer is an abstract class that enables us to have options to view the data. Its own polymorphic displayViewer method had a minor impact on the existing design. In this elaboration phase we implemented the console-viewer only however, extensions required for new variations i.e. (graphic viewer) are easy to add in future iterations since the new implementation can be added without affecting other classes. Therefore, data visualization can occur in many different forms by simply assigning different responsibilities to the UserInterface class using Visualizer. Using the Polymorphism design pattern made class extending to handle new changes much simpler in this iteration.

III. Protected Variations

In general, most design patterns are used to protect software against changes. The Protected Variations design pattern is used to protect classes in the software from potential future changes in classes with numerous variations. In our first elaboration iteration, our code was not protected against variations, as we did not have the necessary knowledge to implement this into our code. However, during the second elaboration phase, our team gained more insight on design patterns and how to effectively use them. We anticipated that the Record class would be very susceptible to changes if the code was used to manipulate a new type of dataset. Therefore, it was very important to create the Template class to be a stable interface that wrapped around the fluctuating Record class. The Template class had the responsibility of being the records' blueprint such that it can be polymorphically extended to the Record class and any changes in it to accommodate a new dataset will not affect any other class. Thus, new implementations of this code to read new datasets can be introduced without affecting other classes since the other elements in the code are excluded from changes.

IV. Creator

The Creator design pattern focuses on keeping the system independent from the creation of its objects. This pattern helped us decide which class should be responsible for creating instances of data visualization options. In our code, using the Creator pattern as a guide, we decided that the Main class should oversee the creation of the *UserInterface* instances or any other future visualization classes. The reason for this is that the Main class can get the initializing information for the user and then create the required visualization instance based on the user's selection. Therefore, only the required instance will be created which will improve the code's performance. Having the Main class be responsible for creating the required visualization mode increased our code's clarity. Implementing the Creator pattern in our code also resulted in lower maintenance time and lower dependency between the classes.

V. High Cohesion

High Cohesion is a design pattern that focuses on how interrelated the components of a program are and how much work each element is doing. In the High Cohesion design pattern, classes are like a collection of elements that share common responsibilities while also being very focused on

their respective tasks. Implementing this design in our code helped us represent our program objects in a manageable and understandable way while maintaining Low Coupling. Every class was assigned specific tasks and relative responsibilities, so that the cohesion level remained high. For example, our Load class only contains attributes and methods that will load the file into the system. The Record class has all the elements related to data structure, record field defining, getting and setting record attributes, etc. Meanwhile, the Template interface is focused on providing the Record class with customized Record specifications. Likewise, the Distance and Findflood classes were centralized to do one task only; the Distance class was used to calculate the distance between addresses and the Findflood class was used to find the flood service request for the required street. Similarly, all the other classes were created to contain elements that serve the same purpose. Therefore, implementing this design kept our program practical, understandable and maintainable, while preserving Low Coupling.

The results we accomplished in this iteration can be divided into two categories: user oriented, and developer oriented. For the users' results, we were able to deliver a working code that displayed the required data based on the user's entry. Whereas for the developers' results, we were able to deliver a code that is highly reusable, fixable and resilient to future changes. It can be easily used to read other datasets simply by making changes to the Template interface only.

In this last phase of our project, the resulting product can be executed from the terminal similar to the previous phase. Instead of entering a filename, it is assumed that the user will have altered the attributes (number of fields, labels, types, filename, delimiter, and primary key field index) in the Template interface to match the dataset they wish to view and interact with. In our product, we kept the two example datasets the same as in previous iterations, the Community Centres and Property Flooding/Grading Service Requests. In this third phase, when the user compiles all the .java files and runs the program on their terminal, they are displayed the number of records loaded from the file and an interactive menu. This menu has more options to view the dataset in different forms. If they choose 1, the first record will be displayed and if they choose 2, the next record will be displayed. Choosing 3 will display the previous record and 4 will display the last record in the dataset. Selecting 5 will display all the records and option 6 will display all the records sorted by the PRIMARY_KEY_FIELD_INDEX in the Template interface. If the user enters 7, they will be prompted to enter the latitude and longitude of an address they want to find the nearest community centres from or enter a street name to get service request history from, depending which API they are in. The program will then display the appropriate response of either a list of community centres with a straight line distance of 5km or less from that address or the service request history for that street, respectively. To exit the program, the user will enter 0. Figures 7-11 in the Appendix provide more details on users' results for this iteration.

10. Discussion

The main objective of this project is to help the people of Windsor make informative decisions when using our APIs. The original assumption was that the APIs should be focused on facilitating access to the datasets, developing useful applications to use the datasets, increasing searchability within them, and providing options for data visualization. To validate the APIs, the

following question was asked, did we build something that captures the city of Windsor stakeholders' needs?

When choosing the Flooding/Grading Services dataset, we had in mind that floods are a common and widespread natural hazard in Windsor. It causes extensive damage to property and results in high property insurance. Our goal is focused on helping new homebuyers/renters to make an informed decision before taking this big step in their lives. When they use the Property Flood/Grading Serves Requests API, they can simply enter the street name and get the history of flood service requests that were received by the City of Windsor for that specific street. This data will help the user decide whether they want to relocate to their chosen property or not. This data could also help insurance companies set their prices according to the number of flood requests in each area.

On the other hand, when we chose the Community Center Locations dataset, our intention was to assist the newcomers to Windsor. Participating in community activities is important as they can help newcomers meet people and adjust to their new life in Windsor. Most statistics show that newcomers rely on public transit, walking or biking, so our Community Center API helps the user to learn about the nearest community center within 5 kilometers from their location.

Therefore, our APIs were validated, as fulfill their intended purpose. They complied with the vision of the City of Windsor to help the stakeholders find information that can be found in the data sets faster.

For API verification, both APIs had to be tested to make sure that they would do what they are supposed to do without any bugs or issues. Both APIs were able to display the required information without any bugs. However, we believe that additional improvements are required to provide more specific information to the user. Due to time constraints and lack of knowledge in some areas, our team was not able to complete some of the requirements we originally planned to have in these APIs. In future interactions, we would like to add a method that gets the user's address or postal code as an input and then displays the nearest community center, adds the google map visualization, and shows the available public transit routes from the user's location to the community center.

As for the flooding services API, we would like to add some graphical charts to show a comparison between different wards and the number of flood requests. And a method to help the user find the streets with the lowest or highest number of flood requests.

11. Conclusion

In conclusion, the objective for this project was to create a platform that makes data easily accessible, dynamic and to create some sort of visualization of that data. Looking back at the eventual findings, the connection that can be made is that the objectives of this project were crucial points since they are the main focus and purpose of any data portal.

What we achieved in the phases of this project are the foundational aspects and building blocks of this platform. Through this project, we managed to load different datasets into a program,

display records of the datasets in different forms, allow the user to interact with the dataset, sort by specific fields, and search through the records to narrow down specific records.for These components are basic requirements that have a variety of applications and uses and are necessary in any data portal.

What we failed to achieve within these first few phases of the project was accessing the data straight from the City of Windsor Open Data Catalogue, so data given to the user would always be real-time and the most up-to-date information. We also failed to create alternate visualizations for the data such as mapping the locations to Google Maps and graphing the data for the user. Lastly, a quality that would've made navigability through the portal even easier would be a "related dataset" feature which would display some datasets that correlated with what the user was currently viewing.

Due to our prototype being a reflection of the basic features an open data portal should have, it has some limitations and assumptions that the user must adhere to in order to get full use of it. In Phase 2, the user must ensure that the datasets being used with the program are specifically the Community Centres and Property Flooding/Grading Service Requests from the City of Windsor Open Data Catalogue or some other dataset that has the exact same format as these two datasets. In Phase 3, we removed this limitation and instead refactored our program so the user would be free to load and access whichever dataset they would like, and would only have to fill in the attributes of that dataset in the Template class.

The two datasets mentioned above are used as a baseline for future work where the team plans to generalize the findings from the first two phases to create an API that is suitable to a variety of datasets. The benefits of our programs are that they allow the user to navigate through the dataset in a very convenient and user-friendly way. Benefits also include automated retrieval and sorting of the datasets, an easy to look at list visualization, and dataset-specific applications that are easy to use and understand.

12. Future Work

Our future plan is to develop this work into a commercial product that can be used to enhance the existing City of Windsor Open Data Catalogue. These APIs have great potential to help make this website more friendly for users and developers. A quick market research shows that the City of Windsor doesn't have an official application that uses these datasets for other applications.

The U.S. The Federal Emergency Management Agency (FEMA) has an easy tool that shows if your address is in a flood zone [7]. This is one of the features that we plan to implement in the future iterations of the Flooding Services API. This information could be very useful for potential residents if they are trying to find a new home in Windsor but would like to make sure their new address is not susceptible to flooding. As of our current research, since Windsor is surrounded by bodies of water, residents of this city have experienced various types of flood emergencies in the past [8]. Since there is no tool like this on the current Open Data Catalogue, adding an application to check if an address is in a flood zone could be widely used and appreciated by the residents of this city.

Due to the current COVID-19 situation, most people are interested in having quick access to finding out if they have been exposed or what areas to avoid to prevent infection. A useful application to help facilitate this process would be to add an alert system that would update the users about any recent positive COVID-19 cases in Community Centres they recently looked up or are in close proximity to. This is just one dataset that could be integrated for this application, however several datasets on the Open Data Catalogue could be used for this as well.

In addition, the software implementation of the APIs in this project is flexible and reusable due to the GRASP designs that were used. Therefore, they can easily adapt to any new dataset and be manipulated in different ways to create applications specific to those datasets.

13. References

[1] GeoDataSource. 2019. *Calculate Distance by Latitude And Longitude Using Java*. [Online] Available: https://www.geodatasource.com/developers/java?fbclid=IwAR0F_3Iu7zhViW7nAtA6EUQcgVdb4Qruv2FPWtxuvSTnyyTiRMpjlt471_U. [Accessed 11 July 2020].

[2] *City of Windsor Opendata*, 2020. [Online]. Available: https://opendata.citywindsor.ca/. [Accessed: 10-Jul-2020].

[3] *City of Toronto Open Data Portal*. [Online]. Available: https://open.toronto.ca/. [Accessed: 10-Jul-2020].

[4] “Access Data and Reports,” *CIHI*. [Online]. Available: https://www.cihi.ca/en/access-data-and-reports. [Accessed: 10-Jul-2020].

[5] "Open Data | Open Government, Government of Canada", *Open Government, Government of Canada,* 2020. [Online]. Available: https://open.canada.ca/en/open-data. [Accessed: 11- Jul- 2020].

[6] “Case Studies,” *ckan*, 07-Jun-2017. [Online]. Available: https://ckan.org/case-studies/. [Accessed: 11-Jul-2020].

[7] "FEMA Flood Map Service Center | Search by Address", *msc.fema.gov*, 2020. [Online]. Available: https://msc.fema.gov/portal/search. [Accessed: 10- Aug- 2020].

[8] "Floods", *citywindsor.ca*, 2020. [Online]. Available:https://www.citywindsor.ca/residents/emergencyandcrimeprevention/Emergency-Preparedness/Floods/Pages/Floods.aspx. [Accessed: 10- Aug- 2020].

14. Appendix A - Group Work

The group collaboration of our team is divided as follows:

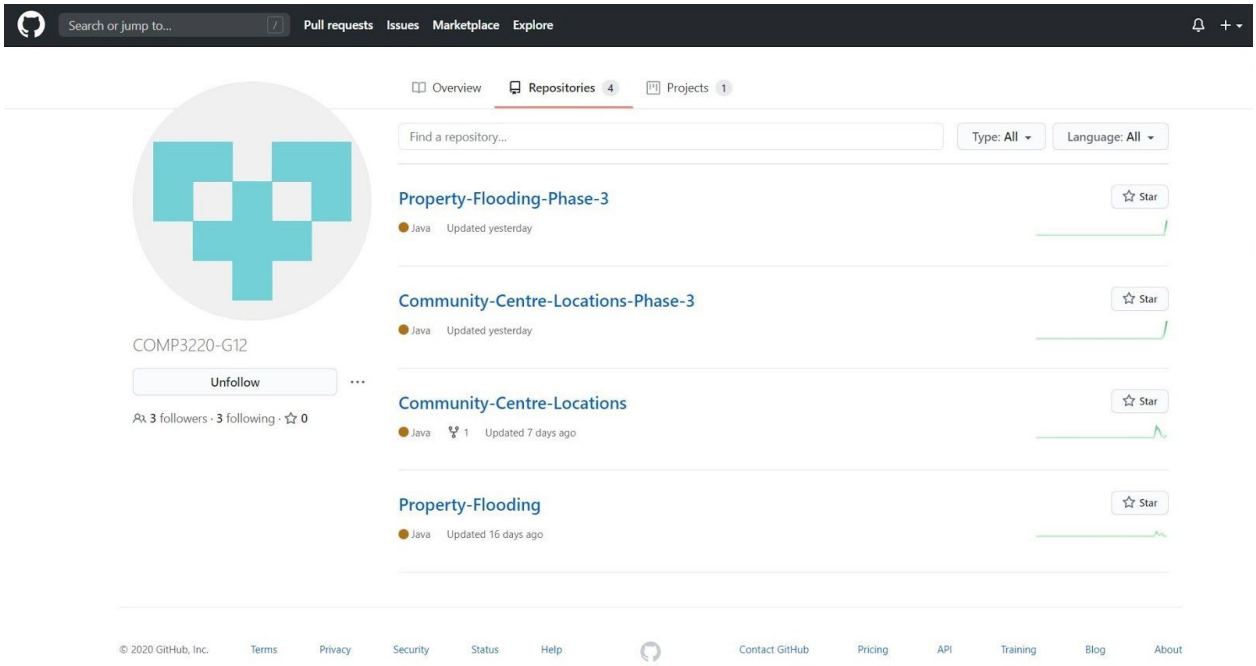
- Lema Khalil - code repository for the flooding dataset, UML diagrams and contribution to the reports.

- Saffa Alvi - code repository for the community centres dataset, UML diagrams, PowerPoint presentation, and contribution to the reports.
- Alina Noor - software testing, myweb management contribution to the report.

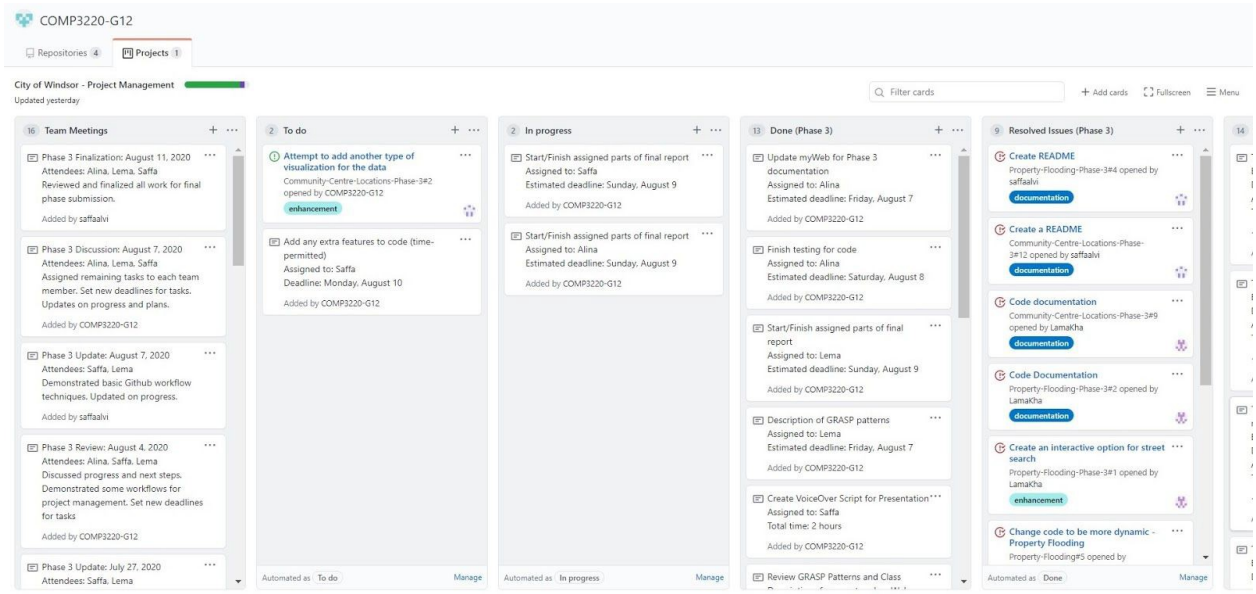
Lastly, the project final report sections were distributed between the team members as follows:

- Saffa: 1, 2, 5, 6, 8, 9, and final review of all sections
- Lama: 3/4, 8, 9, 10, 12, and final review of all sections
- Alina: 7, 11 and 14

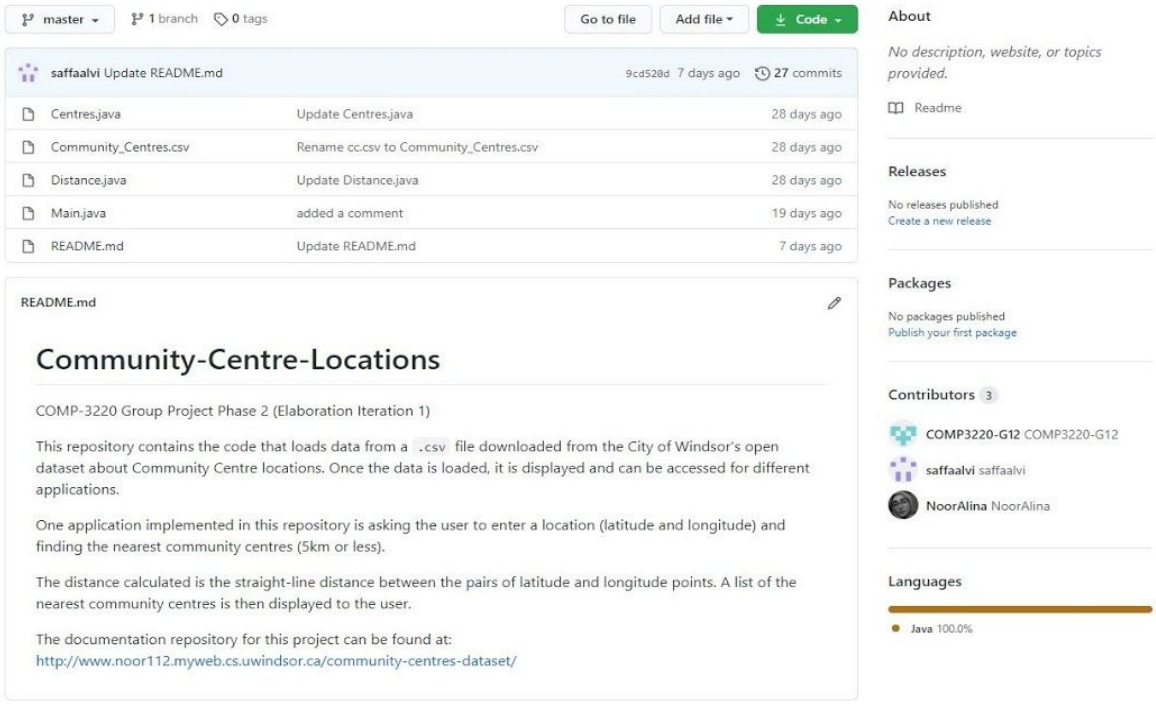
In terms of project management tools, GitHub was used to keep track of all of the tasks in Phases 2 and 3. Tasks were categorized in the form of to do, in progress and done for each phase. For version control, the team used Git and GitHub to update the code as changes were being made. The testing environment used was BlueJ and the testing method was that of unit testing where individual classes were tested to test if they had characteristics of low coupling.



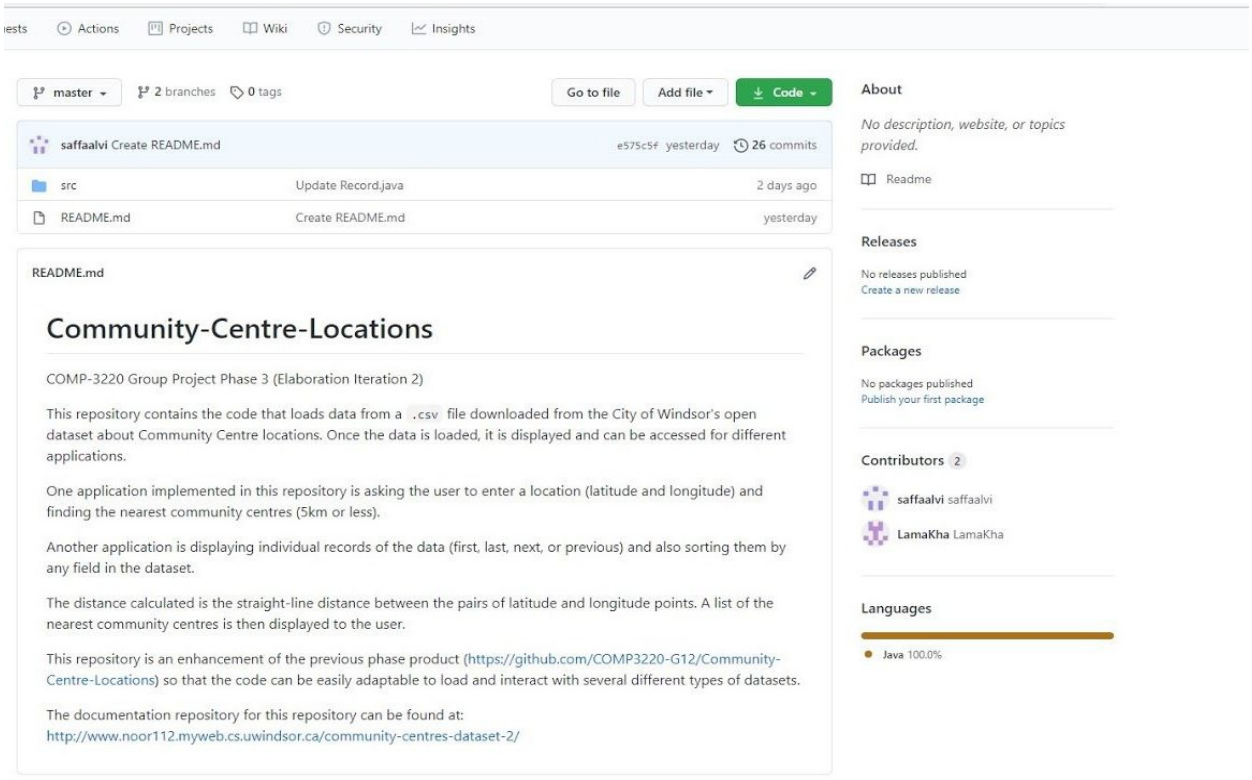
This is an overview of all of the repositories for this project



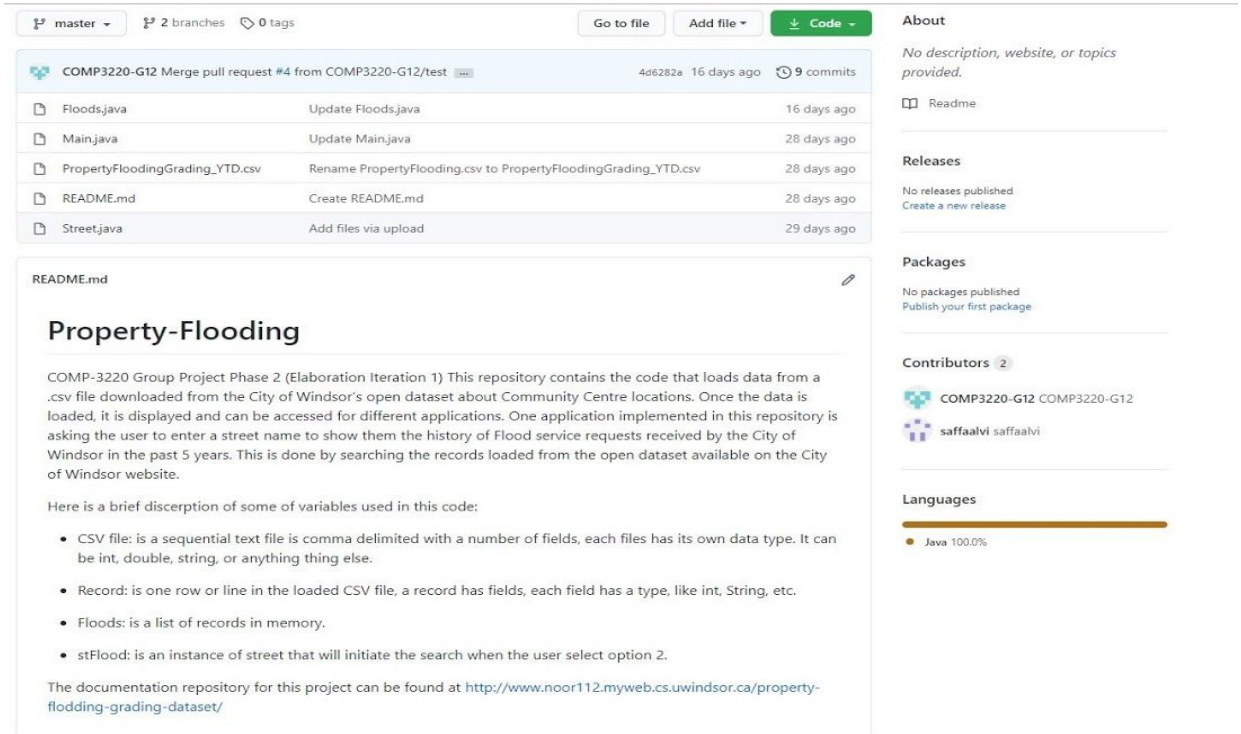
This is the projects management board



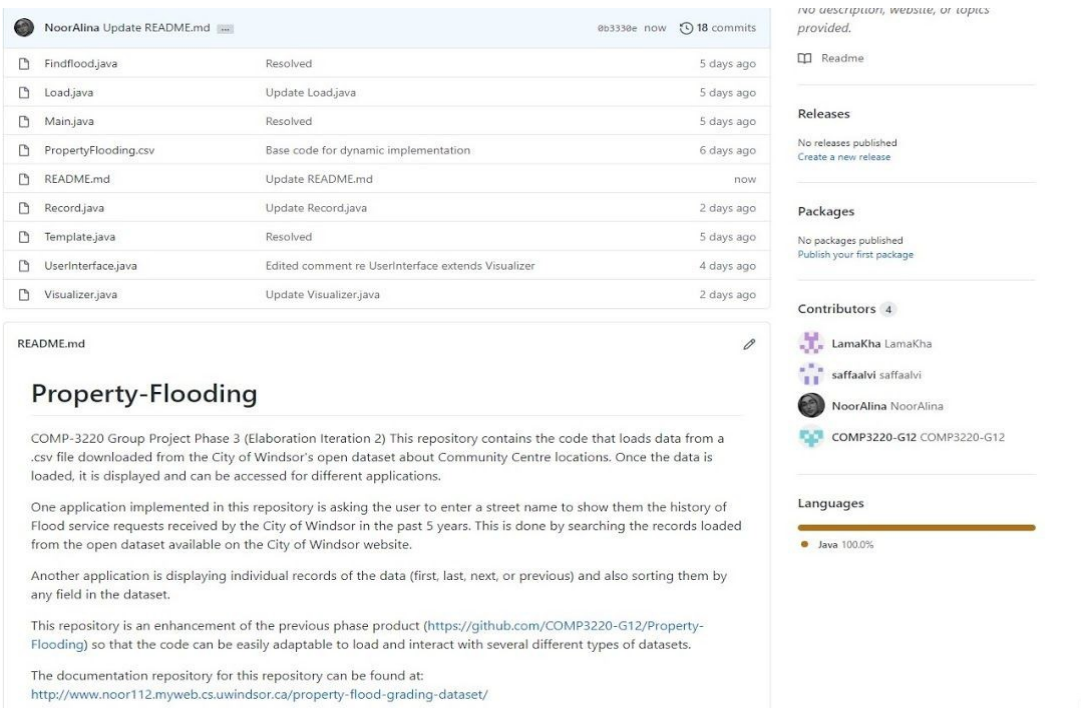
This is the Community Centres Phase 2 repository



This is the Community Centres Phase 3 repository



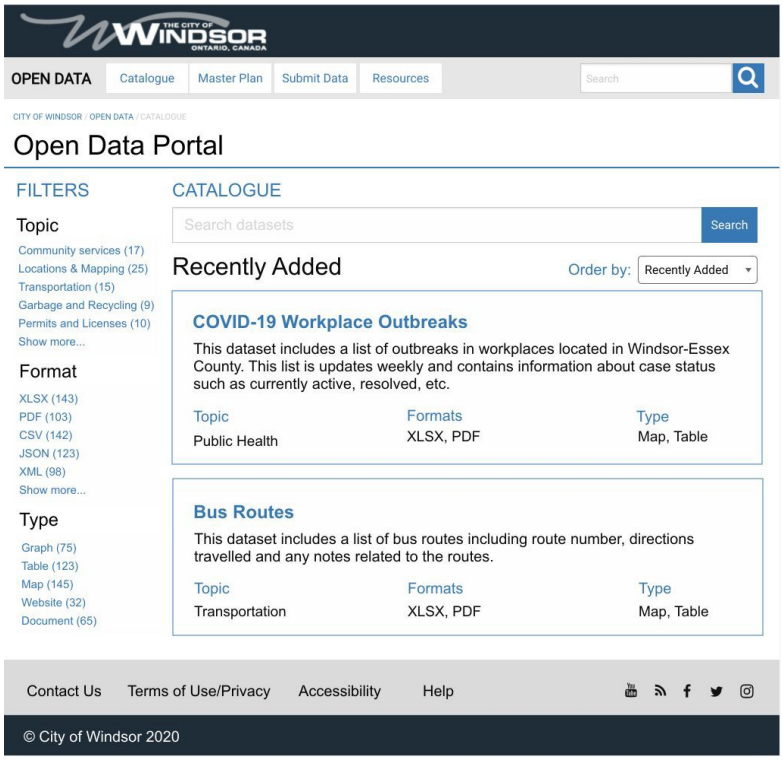
This is the Property Flooding Phase 2 repository

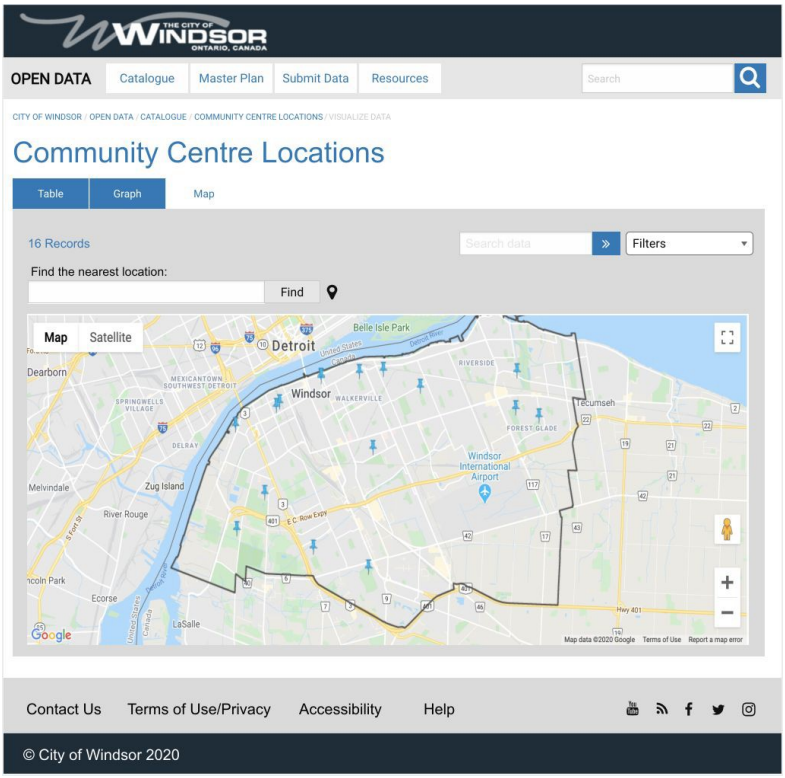


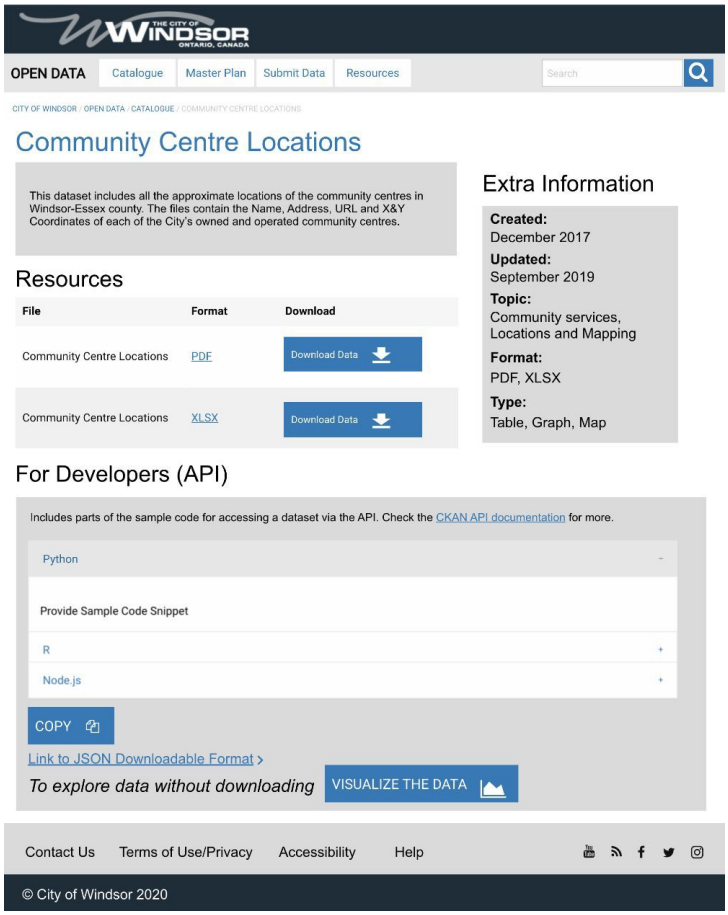
This is the Property Flooding Phase 3 repository

15. Appendix

Figures 1-3: Final Prototype Plan/Vision (Phase 1: Inception)




THE CITY OF WINDSOR ONTARIO, CANADA
OPEN DATA
Catalogue
Master Plan
Submit Data
Resources
Search
CITY OF WINDSOR / OPEN DATA / CATALOGUE / COMMUNITY CENTRE LOCATIONS / VISUALIZE DATA
Community Centre Locations
Table
Graph
Map
16 Records
Search data
Filters
Find the nearest location:
Find
Map
Satellite
Detroit
Belle Isle Park
Windsor
Dearborn
Tecumseh
Windsor International Airport
Zug Island
River Rouge
Melvindale
Ecorse
LaSalle
Google
Contact Us
Terms of Use/Privacy
Accessibility
Help
© City of Windsor 2020


THE CITY OF WINDSOR ONTARIO, CANADA
OPEN DATA
Catalogue
Master Plan
Submit Data
Resources
Search
CITY OF WINDSOR / OPEN DATA / CATALOGUE / COMMUNITY CENTRE LOCATIONS
Community Centre Locations
This dataset includes all the approximate locations of the community centres in Windsor-Essex county. The files contain the Name, Address, URL and X&Y Coordinates of each of the City's owned and operated community centres.
Extra Information
Created:
December 2017
Updated:
September 2019
Topic:
Community services, Locations and Mapping
Format:
PDF, XLSX
Type:
Table, Graph, Map
Resources
File
Format
Download
Community Centre Locations
PDF
Download Data
Community Centre Locations
XLSX
Download Data
For Developers (API)
Includes parts of the sample code for accessing a dataset via the API. Check the CKAN API documentation for more.
Python
Provide Sample Code Snippet
R
Node.js
COPY
Link to JSON Downloadable Format ›
To explore data without downloading
VISUALIZE THE DATA
Contact Us
Terms of Use/Privacy
Accessibility
Help
© City of Windsor 2020

Figures 4-5: Property Flooding/Grading Resulting Product (Phase 2: Elaboration Iteration 1)

```
Saffas-MacBook-Pro:~ saffaalvi$ cd IdeaProjects/Flooding2/src
Saffas-MacBook-Pro:src saffaalvi$ javac *.java
Saffas-MacBook-Pro:src saffaalvi$ java Main
Enter the name of the dataset you would like to view: PropertyFlooding.csv
Number of records loaded: 1038
1. Display All Records.
2. Display Flood Services History using Street Name.
0. Quit
Choose menu item: 2
Welcome to The City of Windsor's Flooding History API
ENTER THE STREET NAME: Wyandotte
Floodings reported at the street entered are:
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-12-05 16:53, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-06 15:12, Number: 70##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-01 11:29, Number: 998##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-15 11:32, Number: 998##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-06-17 15:33, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-06-20 11:12, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-23 10:11, Number: 83##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-01 10:32, Number: 65##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-15 10:30, Number: 65##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-10-16 16:52, Number: 11##, Street: WYANDOTTE, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-03-23 9:35, Number: 50##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-05-15 9:58, Number: 998##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-05-14 10:25, Number: 997##, Street: WYANDOTTE, Ward: WARD 7
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-06-27 15:01, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-09-26 8:58, Number: 58##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2018-05-28 9:00, Number: 998##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2017-01-16 10:18, Number: 14##, Street: WYANDOTTE, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2017-07-13 11:36, Number: 84##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2017-08-29 17:53, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2017-11-06 9:47, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 9/29/2016 10:59, Number: 38##, Street: WYANDOTTE, Ward: WARD 5
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-12-05 16:53, Number: 63##, Street: WYANDOTTE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2019-05-06 15:12, Number: 70##, Street: WYANDOTTE, Ward: WARD 6
```

```
Choose menu item: 1
Displaying all data:
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: E-Mail, Date: 2020-02-24 12:58, Number: 4##, Street: ELINOR, Ward: WARD 7
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-03-30 15:20, Number: 18##, Street: HALL, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-02-14 9:50, Number: 5##, Street: CAMERON, Ward: WARD 2
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 12:08, Number: 17##, Street: CHATEAU, Ward: WARD 7
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 9:57, Number: 11##, Street: ISABELLE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 11:25, Number: 10##, Street: BUCKINGHAM, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 11:24, Number: 10##, Street: PARENT, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-17 12:33, Number: 78##, Street: ST ROSE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-27 12:30, Number: 12##, Street: JANISSE, Ward: WARD 6
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-03-25 10:17, Number: 37##, Street: WOODWARD, Ward: WARD 9
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-03-20 10:35, Number: 20##, Street: ARRAS, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 15:38, Number: 16##, Street: DOUGALL, Ward: WARD 3
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-27 15:13, Number: 4##, Street: CLINTON, Ward: WARD 3
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: E-Mail, Date: 2020-02-04 11:06, Number: 18##, Street: LAMBTON, Ward: WARD 1
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-05-05 9:01, Number: 16##, Street: CLEARWATER, Ward: WARD 7
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-13 14:43, Number: 10##, Street: DOUGALL, Ward: WARD 3
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-22 10:02, Number: 17##, Street: LINCOLN, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-03 13:14, Number: 3##, Street: CHILVER, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-03 13:14, Number: 3##, Street: CHILVER, Ward: WARD 4
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-03-10 15:23, Number: 17##, Street: JEFFERSON, Ward: WARD 8
PG/F: Property Flooding / Grading, Department: Inspections, Contact Method: Phone, Date: 2020-01-20 9:46, Number: 114##, Street: CORMORANT, Ward: WARD 7
```

Figure 6: Community Centers Location Resulting Product (Phase 2: Elaboration Iteration 1)

```
Saffas-MacBook-Pro:~ saffaalvi$ cd IdeaProjects/CityofWindsorPhase2
Saffas-MacBook-Pro:CityofWindsorPhase2 saffaalvi$ cd src
Saffas-MacBook-Pro:src saffaalvi$ javac *.java
Saffas-MacBook-Pro:src saffaalvi$ java Main
Enter the filename: Community_Centres.csv
Number of records loaded: 16
1. Display All Records.
2. Find nearest Community Centre.
0. Quit
Choose menu item: 1
Displaying all the community centres:
FID: 0, Address: 2861 Lauzon Rd, Name: Little River Golf Course, X: -82.9281714155, Y: 42.3060794922
FID: 1, Address: 455 Kennedy Dr W, Name: Roseland Golf and Curling Club, X: -83.0061294912, Y: 42.2512404884
FID: 2, Address: 3215 Forest Glade Dr, Name: Forest Glade Community Centre, X: -82.9157157713, Y: 42.3031557573
FID: 3, Address: 5200 Matchette Rd, Name: Ojibway Nature Centre, X: -83.0757200686, Y: 42.264275747
FID: 4, Address: 2555 Pulford St, Name: Capri Pizzeria Recreation Complex, X: -83.035508601, Y: 42.2579203855
FID: 5, Address: 4270 Alice St, Name: Constable John Atkinson Memorial Community Centre, X: -82.978542886, Y: 42.3134465277
FID: 6, Address: 495 Glengarry Ave, Name: Glengarry Court Community Centre, X: -83.0311238227, Y: 42.3172913011
FID: 7, Address: 1899 Niagara St, Name: Willistead Heritage Complex, X: -83.0111297378, Y: 42.3185618207
FID: 8, Address: 3277 Sandwich St, Name: Mackenzie Hall Cultural Centre, X: -83.0764985562, Y: 42.3001247118
FID: 9, Address: 1551 Wyandotte St W, Name: Adie Knox Herman Recreation Complex, X: -83.0533685861, Y: 42.3079139735
FID: 10, Address: 1168 Drouillard Rd, Name: Gino A. Marcus Community Complex, X: -82.9982774303, Y: 42.3192832921
FID: 11, Address: 400 Wyandotte St E, Name: Windsor Water World, X: -83.0311857907, Y: 42.3166633391
FID: 12, Address: 4200 Malden Rd, Name: Malden Visitors Centre, X: -83.0609758472, Y: 42.2768354264
FID: 13, Address: 1075 Ypres Ave, Name: Optimist Community Centre, X: -83.0036705418, Y: 42.2923614249
FID: 14, Address: 635 McEwan Ave, Name: Centres for Seniors, X: -83.0547864878, Y: 42.3068545485
FID: 15, Address: 8787 McHugh St, Name: WFCU Centre, X: -82.9274864852, Y: 42.3187171903
Choose menu item: 2
Enter latitude of your address: 42.304321
Enter longitude of your address: -83.066040
Community Centres that are 5km or less to your location are:
FID: 3, Address: 5200 Matchette Rd, Name: Ojibway Nature Centre, X: -83.0757200686, Y: 42.264275747, Distance (in km): 4.5232554837019885
FID: 6, Address: 495 Glengarry Ave, Name: Glengarry Court Community Centre, X: -83.0311238227, Y: 42.3172913011, Distance (in km): 3.212849850636001
FID: 7, Address: 1899 Niagara St, Name: Willistead Heritage Complex, X: -83.0111297378, Y: 42.3185618207, Distance (in km): 4.784569830409599
FID: 8, Address: 3277 Sandwich St, Name: Mackenzie Hall Cultural Centre, X: -83.0764985562, Y: 42.3001247118, Distance (in km): 0.9784819085909251
FID: 9, Address: 1551 Wyandotte St W, Name: Adie Knox Herman Recreation Complex, X: -83.0533685861, Y: 42.3079139735, Distance (in km): 1.1159467725231722
FID: 11, Address: 400 Wyandotte St E, Name: Windsor Water World, X: -83.0311857907, Y: 42.3166633391, Distance (in km): 3.1775365211628173
FID: 12, Address: 4200 Malden Rd, Name: Malden Visitors Centre, X: -83.0609758472, Y: 42.2768354264, Distance (in km): 3.0843646132543077
FID: 14, Address: 635 McEwan Ave, Name: Centres for Seniors, X: -83.0547864878, Y: 42.3068545485, Distance (in km): 0.9673259973259245
Choose menu item: 0
Saffas-MacBook-Pro:src saffaalvi$
```

Figures 7-9: Property Flooding/Grading Resulting Product (Phase 3: Elaboration Iteration 2)

```
Saffas-MacBook-Pro:~ saffaalvi$ cd IdeaProjects/Property-Flooding-Phase-3/src
Saffas-MacBook-Pro:src saffaalvi$ javac *.java
Saffas-MacBook-Pro:src saffaalvi$ java Main
Entering console mode...
Number of loaded records: 1038
1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Display Flood Services History using Street Name.
0. Exit
Enter menu option: 6
Records will be sorted by Street
 Request: Property Flooding / Grading, Section: Inspections, Contact: Web Intake, Date: 2017-08-30 15:16, Unit:  , Street:  , Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Web Intake, Date: 2018-06-07 8:38, Unit:  , Street:  , Ward: WARD 1
 Request: Property Flooding / Grading, Section: Inspections, Contact: Web Intake, Date: 2018-06-01 14:00, Unit:  , Street:  , Ward: WARD 1
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-04-07 9:17, Unit: 961##, Street: 42, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-04-07 9:17, Unit: 961##, Street: 42, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-07-12 10:54, Unit: 45##, Street: 8TH CON, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-11-07 8:52, Unit: 45##, Street: 8TH CON, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2016-04-06 15:28, Unit: 12##, Street: ABBEY, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2016-04-20 17:37, Unit: 31##, Street: ACADEMY, Ward: WARD 1
 Request: Property Flooding / Grading, Section: Inspections, Contact: E-Mail, Date: 2016-03-14 11:58, Unit: 35##, Street: ACADEMY, Ward: WARD 1
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-03 10:06, Unit: 28##, Street: ACADEMY, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2014-07-10 16:01, Unit: 12##, Street: AIRE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 10/11/2016 14:05, Unit: 12##, Street: AIRE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2020-03-10 15:36, Unit: 11##, Street: ALBERT, Ward: WARD 5
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-09-26 9:31, Unit: 15##, Street: ALBERT, Ward: WARD 5
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-10-01 12:24, Unit: 15##, Street: ALBERT, Ward: WARD 5
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2016-06-09 10:33, Unit: 15##, Street: ALBERT, Ward: WARD 5
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-02-20 13:18, Unit: 26##, Street: ALEXANDRA, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-02-20 12:50, Unit: 27##, Street: ALEXANDRA, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: E-Mail, Date: 2018-09-24 10:05, Unit: 978##, Street: ALEXANDRA, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-09-17 11:10, Unit: 22##, Street: ALEXANDRA, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-09-05 14:40, Unit: 25##, Street: ALEXANDRA, Ward: WARD 10
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2015-08-18 10:39, Unit: 27##, Street: ALEXANDRA, Ward: WARD 10
```

```
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2016-10-11 8:24, Unit: 39##, Street: ZANZIBAR, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2016-10-11 8:24, Unit: 39##, Street: ZANZIBAR, Ward: WARD 9
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 10/11/2016 8:24, Unit: 39##, Street: ZANZIBAR, Ward: WARD 9

1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Display Flood Services History using Street Name.
0. Exit
Enter menu option: 7
Welcome to The City of Windsor's Flooding History API
ENTER THE STREET NAME: Wyandotte
Floodings reported at the street entered are:
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-12-05 16:53, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-06 15:12, Unit: 70##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-01 11:29, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-15 11:32, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-06-17 15:33, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-06-20 11:12, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-23 10:11, Unit: 83##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-01 10:32, Unit: 65##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-15 10:30, Unit: 65##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-10-16 16:52, Unit: 11##, Street: WYANDOTTE, Ward: WARD 4
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-03-23 9:35, Unit: 50##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-15 9:58, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-14 10:25, Unit: 997##, Street: WYANDOTTE, Ward: WARD 7
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-06-27 15:01, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-09-26 8:58, Unit: 58##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-28 9:00, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-01-16 10:18, Unit: 14##, Street: WYANDOTTE, Ward: WARD 4
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-07-13 11:36, Unit: 84##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-08-29 17:53, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-11-06 9:47, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 9/29/2016 10:59, Unit: 38##, Street: WYANDOTTE, Ward: WARD 5
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-12-05 16:53, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
```

```
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-05-15 10:30, Unit: 65##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2019-10-16 16:52, Unit: 11##, Street: WYANDOTTE, Ward: WARD 4
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-03-23 9:35, Unit: 50##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-15 9:58, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-14 10:25, Unit: 997##, Street: WYANDOTTE, Ward: WARD 7
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-06-27 15:01, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-09-26 8:58, Unit: 58##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2018-05-28 9:00, Unit: 998##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-01-16 10:18, Unit: 14##, Street: WYANDOTTE, Ward: WARD 4
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-07-13 11:36, Unit: 84##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-08-29 17:53, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 2017-11-06 9:47, Unit: 63##, Street: WYANDOTTE, Ward: WARD 6
 Request: Property Flooding / Grading, Section: Inspections, Contact: Phone, Date: 9/29/2016 10:59, Unit: 38##, Street: WYANDOTTE, Ward: WARD 5
1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Display Flood Services History using Street Name.
0. Exit
Enter menu option: 0
Saffas-MacBook-Pro:src saffaalvi$
```

Figures 10-11: Community Centers Location Resulting Product (Phase 3: Elaboration Iteration 2)

```
Saffas-MacBook-Pro:~ saffaalvi$ cd IdeaProjects/Community-Centre-Locations-Phase-3/src
Saffas-MacBook-Pro:src saffaalvi$ javac *.java
Saffas-MacBook-Pro:src saffaalvi$ java Main
Entering console mode...
Number of loaded records: 16
1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Find nearest community centre (straight line distance).
0. Exit
Enter menu option: 6
Records will be sorted by Name
 FID: 9, Address: 1551 Wyandotte St W, Name: Adie Knox Herman Recreation Complex, X: -83.053368586100007, Y: 42.307913973500000
 FID: 4, Address: 2555 Pulford St, Name: Capri Pizzeria Recreation Complex, X: -83.035508601000004, Y: 42.257920385500000
 FID: 14, Address: 635 McEwan Ave, Name: Centres for Seniors, X: -83.054786487800001, Y: 42.306854548499999
 FID: 5, Address: 4270 Alice St, Name: Constable John Atkinson Memorial Community Centre, X: -82.978542886000000, Y: 42.313446527700002
 FID: 2, Address: 3215 Forest Glade Dr, Name: Forest Glade Community Centre, X: -82.915715771300000, Y: 42.303155757299997
 FID: 10, Address: 1168 Drouillard Rd, Name: Gino A. Marcus Community Complex, X: -82.998277430300007, Y: 42.319283292100003
 FID: 6, Address: 495 Glengarry Ave, Name: Glengarry Court Community Centre, X: -83.031123822699996, Y: 42.317291301099999
 FID: 0, Address: 2861 Lauzon Rd, Name: Little River Golf Course, X: -82.928171415500003, Y: 42.306079492199999
 FID: 8, Address: 3277 Sandwich St, Name: Mackenzie Hall Cultural Centre, X: -83.076498556199994, Y: 42.300124711800002
 FID: 12, Address: 4200 Malden Rd, Name: Malden Visitors Centre, X: -83.060975847199998, Y: 42.276835426399998
 FID: 3, Address: 5200 Matchette Rd, Name: Ojibway Nature Centre, X: -83.075720068600006, Y: 42.264275746999999
 FID: 13, Address: 1075 Ypres Ave, Name: Optimist Community Centre, X: -83.003670541800005, Y: 42.292361424900001
 FID: 1, Address: 455 Kennedy Dr W, Name: Roseland Golf and Curling Club, X: -83.006129491199999, Y: 42.251240488400001
 FID: 15, Address: 8787 McHugh St, Name: WFCU Centre, X: -82.927486485200006, Y: 42.318717190299999
 FID: 7, Address: 1899 Niagara St, Name: Willistead Heritage Complex, X: -83.011129737800005, Y: 42.318561820699998
 FID: 11, Address: 400 Wyandotte St E, Name: Windsor Water World, X: -83.031185790699993, Y: 42.316663339100003

1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
```

```
1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Find nearest community centre (straight line distance).
0. Exit
Enter menu option: 7
Enter latitude of your address: 42.304321
Enter longitude of your address: -83.066040
Community Centres that are 5km or less to your location are:
 FID: 9, Address: 1551 Wyandotte St W, Name: Adie Knox Herman Recreation Complex, X: -83.053368586100007, Y: 42.307913973500000, Distance: 1.11594677252317
22 km
 FID: 14, Address: 635 McEwan Ave, Name: Centres for Seniors, X: -83.054786487800001, Y: 42.306854548499999, Distance: 0.9673259973259245 km
 FID: 6, Address: 495 Glengarry Ave, Name: Glengarry Court Community Centre, X: -83.031123822699996, Y: 42.317291301099999, Distance: 3.212849850636001 km
 FID: 8, Address: 3277 Sandwich St, Name: Mackenzie Hall Cultural Centre, X: -83.076498556199994, Y: 42.300124711800002, Distance: 0.9784819085909251 km
 FID: 12, Address: 4200 Malden Rd, Name: Malden Visitors Centre, X: -83.060975847199998, Y: 42.276835426399998, Distance: 3.0843646132543077 km
 FID: 3, Address: 5200 Matchette Rd, Name: Ojibway Nature Centre, X: -83.075720068600006, Y: 42.264275746999999, Distance: 4.5232554837019885 km
 FID: 7, Address: 1899 Niagara St, Name: Willistead Heritage Complex, X: -83.011129737800005, Y: 42.318561820699998, Distance: 4.784569830409599 km
 FID: 11, Address: 400 Wyandotte St E, Name: Windsor Water World, X: -83.031185790699993, Y: 42.316663339100003, Distance: 3.1775365211628173 km
1. Display first record.
2. Display next record.
3. Display previous record.
4. Display last record.
5. Display all records.
6. Display sorted records.
7. Find nearest community centre (straight line distance).
0. Exit
Enter menu option: 0
Saffas-MacBook-Pro:src saffaalvi$
```

16. Special Acknowledgement

We would like to acknowledge the School of Computer Science at the University of Windsor for providing students with the virtual hosting environment, MyWeb, which we used to create our documentation repository for this project. This platform was easy to maintain, access, and update with information regarding our project.

We would also like to acknowledge the Microsoft Corporation for developing an effective project management and code repository platform, GitHub. After unsuccessfully experimenting with Redmine, licensed under GNU General Public License v2 (GPL), in the beginning of our

second phase, we decided to switch over to GitHub to track our project management. After creating the project on Redmine, the Redmine on the CS server got corrupted and the functionalities such as adding issues, accessing the repository, document etc. were no longer working. After getting in contact with the Systems Analyst, we decided it was unreliable to work with Redmine in case this occurs closer to the deadline so we chose to track our project management on GitHub. Our project management on GitHub was tracked and used effortlessly as the platform was easily navigable and made collaboration between the team members very convenient.

Lastly, we would like to recognize the software company, JetBrains, for creating a powerful IDE that we used to develop and test the working code for this project; IntelliJ. This IDE came equipped with all the capabilities we needed to properly debug, test, and refactor our working code implementation.