Final Technical Report

Group 12

Ryan Raffoul (raffoulr@uwindsor.ca)

Donovan Longo (longo114@uwindsor.ca)

November 27, 2020

The objective of this project was to create an open data source portal for the City of Windsor. The main goal was to optimize the user experience as much as possible. This was done in hopes that a more seamless user experience leads to more people using the portal and creating projects with it. Throughout phases there were different experiments on how certain features would work. One example of this is using a Java GUI to create a form that would be used to upload data to the portal. Utilizing Java, Group 12 was able to produce a command-line interface portal that would allow users to view data by category, search for category, upload, and manage data. The results display the progress we made through every phase, reaching closer and closer to our end goal.

Keywords

1. User Experience
2. Open Data
3. Easily Accessible
4. Text-Based
5. Experimental

Acknowledgement Section

We would like to acknowledge Donovan Longo for being responsible for: Final Technical Report, DataEntryTestCase2, DataEntryTestCase1, DataManipulator UML diagram, DataEntry UML diagram, sample Data Entry Form, and developing DataEntry.java.

We would like to acknowledge Ryan Raffoul for being responsible for: Inception Phase Report, Elaboration Phase 1 Report, Test Cases for Elaboration 2, CategoryDisplay UML diagram, sample page design for log in, and developing Dataset.java, FileManipulation.java, Tester.java, and the category classes (Arena.java, Category.java, CommunityCentre.java, FireStation.java, Hospital.java, Library.java, Park.java, PoliceStation.java, and School.java).

Introduction

In 2019 Windsor Star published an article stating Windsor tied for the third fastest growing city of Canada [1]. Growing population comes hand in hand with growing needs/requirements for the residents of the city. One of these requirements is giving people the opportunity to view, manage, and utilize data of the city. This can be beneficial to the city because this then gives developers a chance to create apps or websites using this accessible data. With more apps/websites utilizing City of Windsor's data, the information becomes conveniently usable for most residents with smart devices. Currently the city of Windsor has an

Open Data Catalogue that allows anyone to access various data. For example, this includes Arenas, Community Centres, Election Results, Fire Stations and the list goes on [2]. This is where our project comes in. Our biggest issue with the way Windsor's open data catalogue is set up is the user experience. You can only sort resources by their file type or search for specific data using a search bar. Also, when accessing the site on a smart phone, the site is functional but does not perform optimally. One example of this is words going off screen where a user would be having to scroll horizontally to read. With our project we planned on creating a different open data source for the city of Windsor with the focus of user experience. Our goal with this project is that a more seamless interaction between a user and the data source will encourage more people to utilize the data whether it be for reports, articles, or apps. We hope to achieve this because with the products created from these users will help shape the future development of the City of Windsor. Coming together as a community to help the development of Windsor following its slogan: Dream, Dare, Do is our motivation behind this project.

Within this report there are 6 main subjects. Related Work covers projects that we have researched and used that have a similar goal to our own. These pages/projects aided us in gaining more of an idea on how to execute our plan. The Approach will come next, and this covers how we tackled our projects including our thought process, initial protypes and environments, languages, etc. Next in the Experimental setup/Demonstration covers how we built our prototype, the features it includes, as well as how to use it. Then the Discussion section will go over challenges we faced, changes we had to make between phases, and what our results mean. The fifth section is the Conclusion which will comprise on our reflection of our goals; which ones we met, which ones we failed to meet. The Conclusion will also cover the limitations and benefits of our prototype. Lastly is Future Work where the report will describe

where we want to take our product in the near future. After Future Work, there are short categories covering references in this report, appendices, and special acknowledgments.

Related Work

There were a couple resources that our group used to help us come up with our attack plan on this project. Firstly, the City of Windsor's open data catalogue. This web page gave us a perception on how to NOT organize our data. Some elements we took from the resource was the lack of managing data available. We noticed the user interaction with the data is very limited on the site and was the main aspect we wanted to improve upon. One positive element we got from the open data catalogue was the use of google maps when accessing certain data [2]. Another resource we used is Ontario's open data source website. We had a much more pleasant user experience on this site, where everything seemed to flow nicely, and all pages seemed interconnected which is useful for easy navigation. The 2 main takeaways from this resource was how the site separated the data into different categories such as; Agriculture and Food, Health, Economy and Business [3]. The second was their "Order By" feature [3]. This allowed the user to order the data by Relevance, Name ascending/descending, last modified [3]. The reason we took an interest to these features is because they both relate to our main goal of this project, user experience. CKAN was another great resource we read into before starting our project. Going into this project our group had very little to no experience developing a site responsible for managing large amounts of data. CKAN is a known organization that specializes in data platforms that allow people to collect, manage and distribute data to the public [4]. Our group figured that with CKAN we could make our goals for this project achievable. As it being a free open source software, we read into how other sites use it, as well as the customizability that comes with it.

All three of these resources gave us a better understanding of how to set up our project development and key features we wanted to add.

Approach

Our group has the most experience programming with Java, so that is what we decided to utilize going forward with this project. It also ensured that we could apply object-oriented programming in our project. In our inception phase we developed a sample Dataset class. This was our first potential class organizing how each dataset would be stored. It included eight variables to describe the data; dataNumber (Used as and ID number for the datasets), dataTitle, dataDescription, dataPublisher, dataDate, dataImage, datafile, and dataContact. The class consisted of a constructor who would set all the variables from the parameters and get/set methods for each variable. Our thought process behind this was that in the future this would be useful when we need to edit a certain dataset or sort data. Moving forward into our first elaboration phase we designed our first data entry class. Keeping GRASP in mind we wanted to ensure that the class dataset would have a creator which was our DataEntry class. An illustration of both classes is shown in the Appendix B portion of the report (Fig 1). DataEntry was designed as a GUI even though we knew we were going to migrate to HTML and JS on our portal. The GUI would give us the ability to trouble shoot and test our dataEntry class early on in development so we could catch any errors we were missing in our code. This class included eleven variables; frame, dataTitle, dataDesc, publisher, date, dataImg, dataFile, contactInfo, verified, DATA, and counter. Besides the constructor and initializeVariablesOnFrame there was one other method called verification. This method is responsible for verifying that a user completed the data entry form completely so no variables are left as null. We knew we had to communicate to the user to fill out all variables so if they tried submitting with one blank, a pop up would prompt them to go back and fill out the blank variables. With the future elaboration

in mind we also designed a sample UML diagram for a DataManipulator class. Again, keeping GRASP in mind, we thought this class would be a good way of spreading responsibility across classes. This class had methods of sorting and searching through the dataset's fields. An image of the UML diagram for this class is included in Appendix B (Fig 2). In our second elaboration phase we quickly realised we did not have the skillset yet to achieve our initial goal to migrate this into HTML/JS for our front end and Java as our back end. To counter this, we developed a command-line interface that would allow a user to view a category list, search for a category, view data within a specific category, upload data, and manage data. We included 8 categories in this prototype; Arenas, Community Centres, Fire Stations, Libraries, Parks, Police Stations, Fire Stations, Hospitals, and Schools. Each of these categories were developed into their own classes with similar fields and methods from our first dataSet class. These classes are all directly related to a "father" class called Category. Category includes the usual get and set methods but also includes data manipulation methods such as sortAsc(), sortDesc(), and findMostPopular(). The other class we needed to design for this command-line interface was FileManipulation which would be used to read CSV files and set/get file categories. An image of these classes UML diagrams and relations are included in Appendix B (Fig 3). For our prototype we have a tester class as our main class that everything is connected to. This class is responsible for having objects of each category and displaying our menu to the user. Since we are using a command-line interface we believed the easiest way for a user to interact with it would be through numbered options. To prevent errors, if a user inputs an invalid integer, a message will be prompted saying their input was invalid, try again. The tester class utilizes the methods printAll() from our categories when listing the data to the user. It also has a readFile() method that reads and parses data from a csv file into variables. We wanted to include this method because we know the world commonly uses CSV files for data.

Experimental Setup / Demonstration

To launch our Tester, we need to compile 11 java files first. These are our categories, the category.java, FileManipulation.java, as well as our Tester.java. The Tester.java acts as our main class that everything is connected to. After launching the tester class a user is greeted with a welcome message stating “Hello! Welcome to the City of Windsor Open Data Portal (Command Line Version)”. Under this message the user is displayed the Main Menu. The main menu consists of 5 options; Category list, Search for a Category, Upload Data, Manage Data, and Exit. Each of these options are numbered. Below the Exit option the message “Select an Option (by number):” is displayed. This makes it easy for the user to understand how to work our command line menu. Currently in this phase our prototype does not have the features Upload Data and Manage Data implemented yet. This being said a user can select either 1 – Will display all implemented Categories at the time to chose from, 2 – Will let the user input text to search for a category or 5 – Will exit the application. If a user selects inputs 1 (Category List), Arenas, Community Centers, Fire Stations, Hospitals, Libraries, Parks, Police Stations, Schools, and a Back option will all be displayed with numbers beside them. Again, the message “Select an Option (by number):” is display to the user as well. If the user then selects a category, the fields of that category will be listed horizontally across the top and a list of data filling those fields will be listed below. An example of this is included in Appendix C (Fig 4). After the data is displayed the category list is then displayed again. From here the user can select another category to view or select back to return to the main menu. In the main menu, if a user inputs 2 (Search for a Category), the message “ENTER CATEGORY NAME:” is then displayed. From here the user will enter any category name. If it is a valid category, the data from the inputted category will be displayed in the same manner as it was previously. A visual representation of this can be found in Appendix C (Fig 5). The main menu is then displayed under the data shown.

From here the user can continue to view data or exit by entering 5. If a chooses to exit, the message “EXITING…” will be shown.

Discussion

Initially moving from our Inception phase to our first elaboration phase, there was not many changes. We extended onto what we developed in our inception phase by creating new classes that integrated with our original classes. Moving from our first elaboration phase to our second elaboration phase we had more changes than initially expected. We realized that if we want our project to have strong scalability, we had to change around our classes. One example of this is when we were developing the category classes, we built off our initial dataSet class but made necessary changes we discovered. Some of these changes include altering variables (adding latitude and longitude) or adding methods (printAll()). Our prototypes produced predictable behaviour throughout our different phases. This is credited to how we developed and tested our code. When a group member was developing a java class, we both made sure it was “runnable” throughout the development. This allowed us to frequently test our code as we added new variables/methods. We wanted to be thorough while developing so that down the road we would not come across weird errors or results when testing final versions of our prototypes. To add another layer of confidence in our code, each group member reviewed code as it was added to our repository verifying that it covered the requirements, we made for ourselves. A big contribution to validating our code was having group members try to run and use our prototypes without looking at the code first. This allowed us to get the perspective of a first-time user. This was valuable information to use because it gave us the ability to gain a users initial thought process when first being exposed to our prototypes. Since one of our biggest goals of this project was user experience, this information granted us a good understanding on how well we did.

Conclusion

Comparing our initial objectives with our final prototype in our most recent phase although we did not achieve all our goals yet, we were able to create a user-friendly data portal for users. In this phase some limitations of our prototype are that we need separate CSV files per category for us to implement them into our prototype. Another limitation of our prototype is that it is limited to a command-line based interface. Originally, we wanted our data portal to be a web-based portal but because of lack of time, and lack of prior knowledge we could not achieve this during this phase. The major factor contributing to this was the need of learning new languages and skills to be able to implement a java backend with a HTML/CSS/JS front end. Currently our prototype runs with the assumption that the CSV files are all formatted the same, with the fields being ordered; XCoord, YCoord, FID, NAME, ADDRESS, FACILITYID. One last limitation our final prototype is the use of Upload and Manage data. The features were unfortunately unable to be achieved during this phase. Although we had a rough idea on how we wanted to upload data, we were unable to implement it in conjunction with our CSV files without quirky bugs. The group decided it was better to scrap the feature in this phase and implement it in future phases so that we have a functional prototype with no bugs. We were able to achieve implementing GRASP patterns such as Creator and high cohesion through our phases. In elaboration 1, our creator was the dataEntry class, in elaboration 2, our creator is the Tester class. The major benefit we pride ourselves on is that our prototype has a very easy, clear, and concise user experience. All possible options are displayed to a user and there are measures in place to redirect them into the right direction if they enter an invalid input.

Future Work

In the next iteration 2 major requirements we would fulfil is the implementation of Upload and Manage within our command-line interface. The second being the start of

migrating our command-line prototype to a web-based application. To accomplish these requirements, we will need to continue to expand our group skillsets and learn more about web application development. If our group were to eventually be at a stage to release this commercially, we would want to make this resource free to the public. Related back to the introduction, the more people that can utilize our project to create anything, whether it be a mobile app, or an article covering a problem in Windsor the better. With more and more people creating, the more it contributes to the future of Windsor.

References

[1] D. Dave Battagello, "Windsor tied for third fastest-growing city in latest StatsCan census data," *windsorstar*, 29-Mar-2019. [Online]. Available: https://windsorstar.com/news/local-news/windsor-third-fastest-growing-city-in-latest-census. [Accessed: 29-Nov-2020].

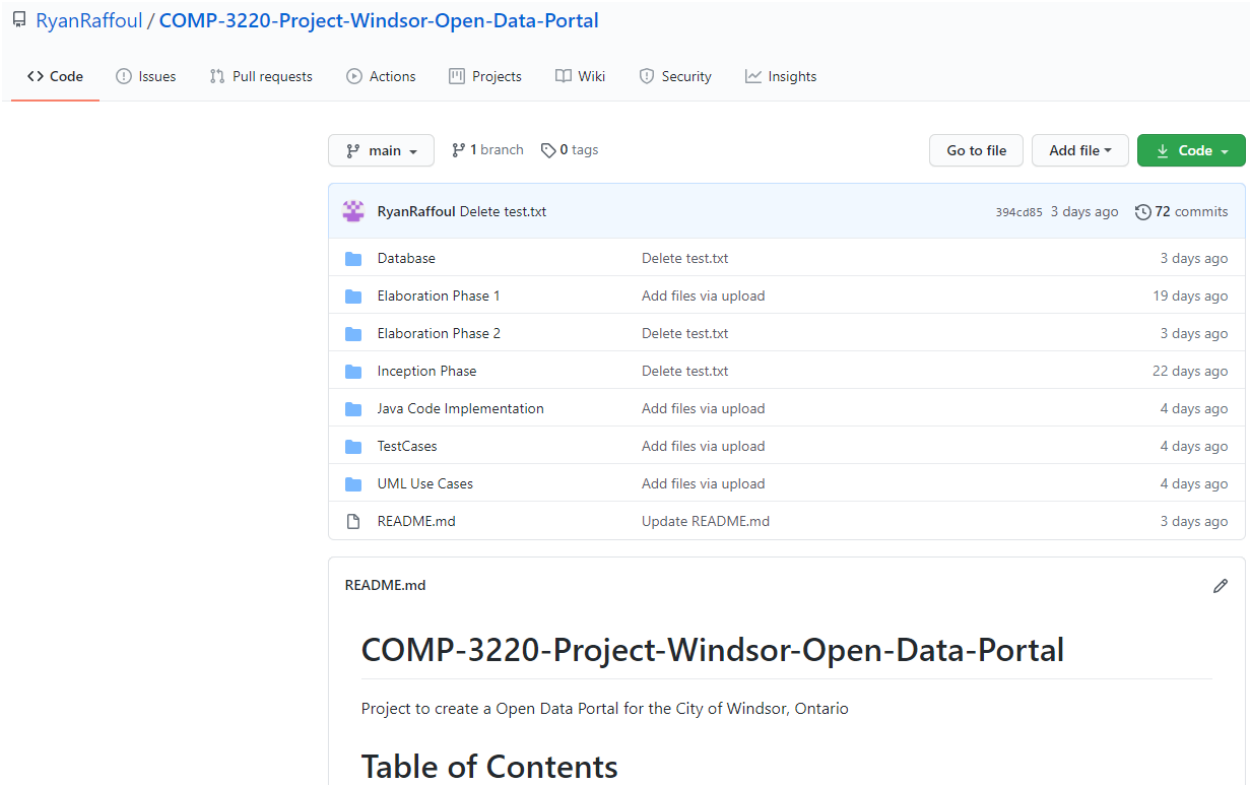
[2] *City of Windsor Opendata*. [Online]. Available: https://opendata.citywindsor.ca/. [Accessed: 29-Nov-2020].

[3] "Home," *Welcome - Ontario Data Catalogue*, 28-Nov-2020. [Online]. Available: https://data.ontario.ca/en/. [Accessed: 29-Nov-2020].

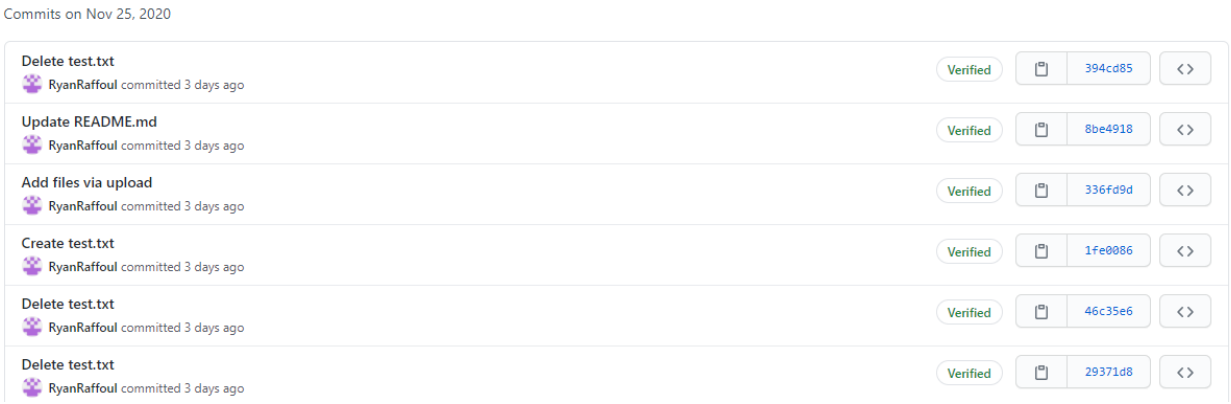
[4] "About," *ckan*, 06-Sep-2017. [Online]. Available: https://ckan.org/about/. [Accessed: 29-Nov-2020].

Appendix A – Group Work

For our group collaboration we used many tools GitHub offered such as a code repository that can be seen in the image below.



To keep track of software version control we utilized GitHub's commits feature. This allowed us to see when a group member added or updated a file in our project. An example of this is seen below. We also would let each other now through texts of any progress we made.

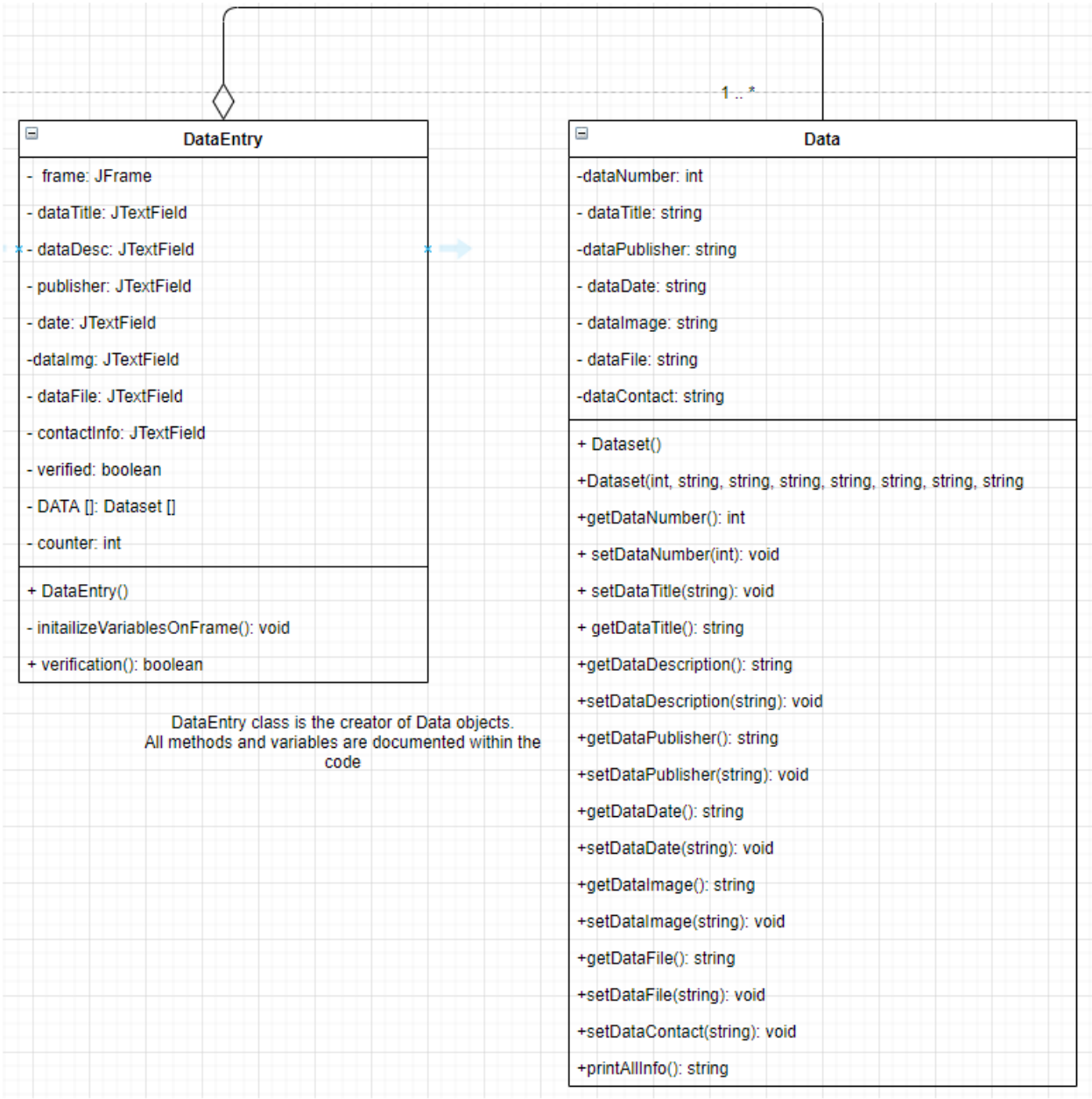


Our testing was done individually on our own workstations when developing. As discussed in the report we also tested and ran each others code to 1: ensure everything ran smoothly, 2: Capture how a users first time experience with our prototypes might go. For time management we utilized a shared Calendar where if one of us inputted an event on a day, we would both see

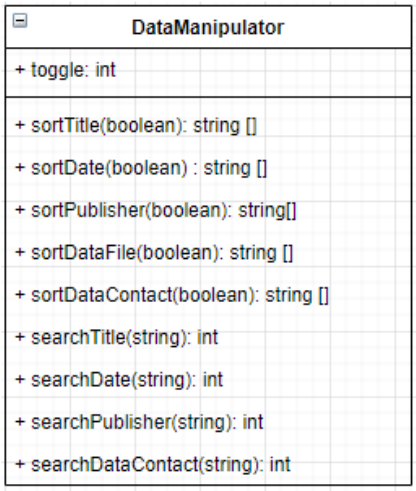
it. Our GitHub repository can be found at: https://github.com/RyanRaffoul/COMP-3220-Project-Windsor-Open-Data-Portal

Appendix B - Approach Details

UML Diagrams for DataEntry and DataSet classes (Fig 1)

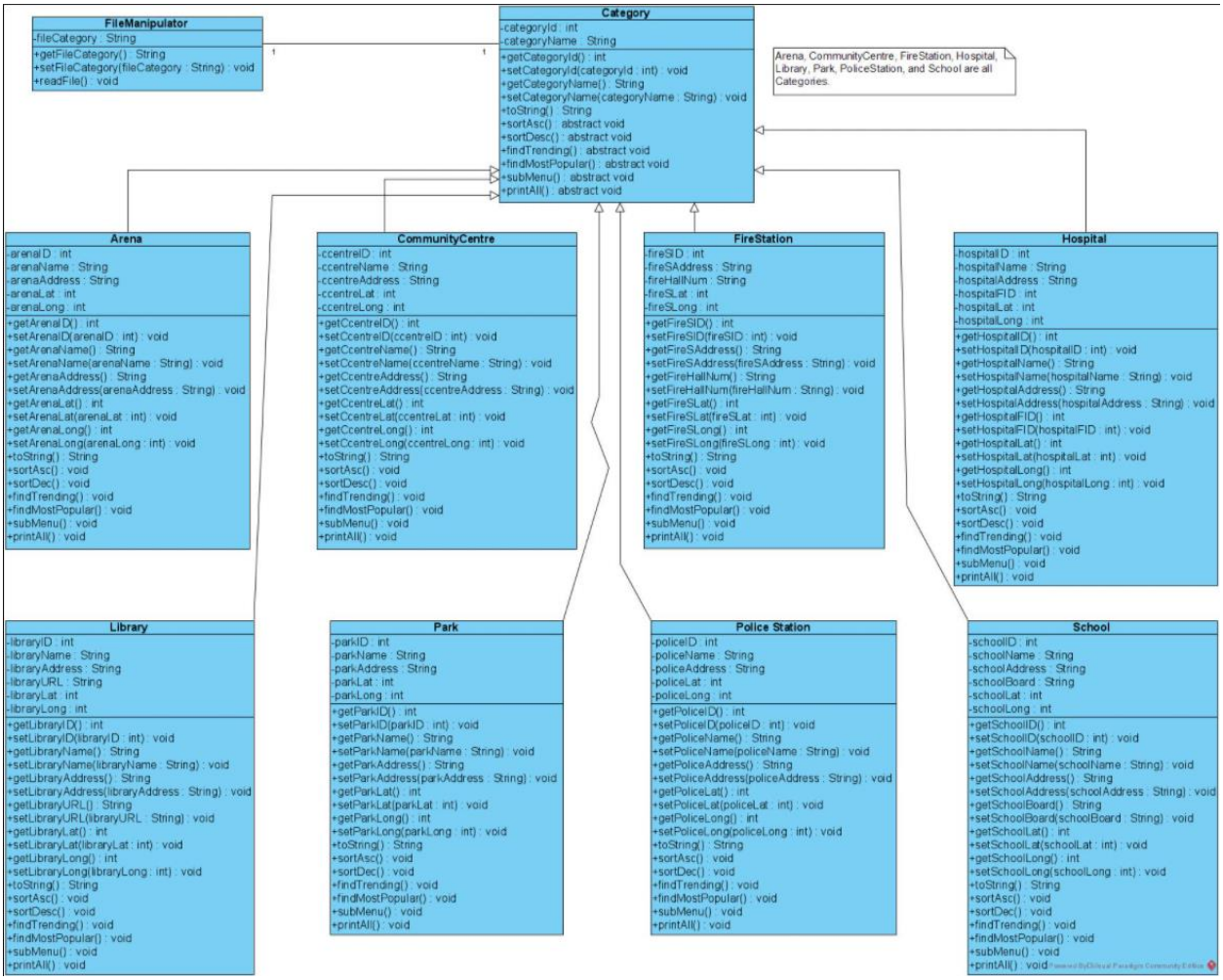


UML Diagram for DataManipulator class (Fig 2)



This is a sample class that would be in charge of sorting and searching Datasets. The sort methods have a Boolean parameter which depicts if the data is sorting ascending or descending. They will then return a list of the specified attribute in a string array. The search methods will iterate through the Datasets looking for the inputted string. Once it has found it, it will return an int id of where it is located in the Data array. If it is not found, a negative number is returned. This class will allow us to spread responsibility across classes following GRASP patterns.

UML diagrams of classes utilized in Elaboration 2. A more clear image can be found in the GitHub repository (Fig 3).



Appendix C – Demonstration Details

Data being displayed to the user (Fig 4).

```
Select an Option (by number): 2
COMMUNITY CENTRE ID  COMMUNITY CENTRE NAME  COMMUNITY CENTRE ADDRESS  COMMUNITY CENTRE LATITIUDE  COMMUNITY CENTRE LONGITUDE
0  Little River Golf Course  2861 Lauzon Rd  -82.92817  42.30608
1  Roseland Golf and Curling Club  455 Kennedy Dr W  -83.00613  42.25124
2  Forest Glade Community Centre  3215 Forest Glade Dr  -82.91572  42.303154
3  Ojibway Nature Centre  5200 Matchette Rd  -83.07572  42.264275
4  Capri Pizzeria Recreation Complex  2555 Pulford St  -83.03551  42.25792
5  Constable John Atkinson Memorial Community Centre  4270 Alice St  -82.978546  42.313446
6  Glengarry Court Community Centre  495 Glengarry Ave  -83.03112  42.31729
7  Willistead Heritage Complex  1899 Niagara St  -83.01113  42.31856
8  Mackenzie Hall Cultural Centre  3277 Sandwich St  -83.0765  42.300125
9  Adie Knox Herman Recreation Complex  1551 Wyandotte St W  -83.05337  42.307915
10  Gino A. Marcus Community Complex  1168 Drouillard Rd  -82.998276  42.319283
11  Windsor Water World  400 Wyandotte St E  -83.03119  42.31666
12  Malden Visitors Centre  4200 Malden Rd  -83.060974  42.276836
13  Optimist Community Centre  1075 Ypres Ave  -83.00367  42.292362
14  Centres for Seniors  635 McEwan Ave  -83.05479  42.306854
15  WFCU Centre  8787 McHugh St  -82.92749  42.318718
```