Final Technical Report

For the Canadian Institute of Health Information (CIHI)

Created by: Dinith Atukorala, John Diaz, Carter Dyck, Saleena Farrukh, Htoo Myat Soe Paing, Keagan Whiston

Due Date: April 07, 2024

Table of Contents

Team Members and Contributions

This is a technical document highlighting the design and basic implementation of a technological dashboard for the **Canadian Institute of Health Information**. The project was developed and executed by **Group 1**, which comprised of the following members:

- Dinith Atukorola (atukorad@uwindsor.ca)
- John Diaz (diaz6@uwindsor.ca)
- Carter Dyck (dyck61@uwindsor.ca)
- Saleena Farrukh (farrukhs@uwindsor.ca)
- Htoo Myat Soe Paing (paing@uwindsor.ca)
- Keagan Whiston (whistonk@uwindsor.ca)

Abstract

As what was briefly stated above, the goal of this project was to design and implement a technological dashboard for the **Canadian Institute of Health Information**. It was designed to allow data visualization and to help draw trends in information for use by healthcare management. The main objectives were creating a user-friendly interface that could seamlessly be integrated with the existing system and also include and display data analytics. Throughout the design process, our team used a mixture of qualitative and quantitative methodologies to ensure the effectiveness of the dashboard. Some key experiments that were used were stress testing and user experience testing. These results helped sculpt the creation of the current product.

Keywords

Below is a list of relevant keywords that were closely tied to the development of the dashboard for the Canadian Institute of Health Information:

1. **Data Analyzation** - The process of examining, transforming and modeling data to find trends and affect future decision-making.
2. **Clinical Insights** - Important findings that are related to clinical practices based on analyzing health data.
3. **Healthcare Information** - The intersection of computer science, information science and healthcare highlighting the methods and resources required to optimize the creation, retrieval and storage of information in the medical field.
4. **Data Visualization** - The process of representing data in different formats such as graphical or visual usually to make more complex data usable and understandable.
5. **Healthcare Trends** - Examining the patterns at large in healthcare, such as the commonness of certain diseases or the delivery of certain healthcare models.

Acknowledgements

Firstly, we want to acknowledge the owner of the dataset of the **Canadian Institute of Health Information** for providing us access to various healthcare data. This project would not have been possible without the continuous work and support from **Group 1**'s combined efforts. Please find more below on specific contributions from certain members:

- **Dinith Atukorola**: For his work on crafting the risk list and management plan, ensuring potential risks are identified and mitigated effectively, as well as contributing to uml diagrams and models.
- **John Diaz**: For developing the comprehensive phase plan and software development plan, providing a roadmap for the project's execution.
- **Carter Dyck**: For his role in implementing core functionalities within the healthcare management tool, laying the foundation for its functionality and usability.
- **Saleena Farrukh**: For her efforts in providing key ideas on development ideas and contributing to the aspects of each iteration phases.
- **Htoo Myat Soe Paing**: For contributions to the use case model and class diagrams, providing essential structures for the project's implementation.
- **Keagan Whiston**: For taking charge of organizing team meetings and ensuring the project maintains high-quality standards before proceeding to the next iterations.

Introduction

Problem statement

The healthcare industry generates massive amounts of data, including patient information, disease patterns, and treatment outcomes, clinical research and more. Accessing, organizing and extracting insights from this data is important for this industry. However, effectively organizing and interpreting this data poses major challenges. There is a need for a comprehensive data management that can effectively hold large amounts of data, while also successfully handling different healthcare datasets, allowing for data retrieval, analysis, and visualization that facilitate decision-making and better patient outcomes.

Motivation

The motivation behind this project is rooted by the crucial role of healthcare data in delivering evidence based data of patients, diseases from the past and discovering future trends in viruses and improving the outcomes and the healthcare system. By creating this comprehensive data management tool, our aim is to make sure that the healthcare professionals, analysts, and researchers can analyze and access healthcare data with the most accurate efficiency effectively. We believe that this tool has the potential to revolutionize the healthcare industry by providing insights on future disease trends, treatment efficiency and budget allocations.

Background

In recent years, advances in medical technology and data collection technologies have resulted in an exponential increase in healthcare data. While there is an abundance of data available, the lack of standardized data management tools and platforms makes it difficult to effectively use this information. These existing data management tools provide accurate answers, but lack the flexibility, scalability and usability aspects, which limits their usefulness in the real-world situations of the healthcare industry, which is to get data fast and accurately while also filtering out a bunch of information when necessary. Our project aims to address and tackle these restrictions by creating a comprehensive, user-friendly data management tool platform which is tailored to the needs of professional healthcare professionals and researchers alike.

Objectives

The main objective of this project is to develop a comprehensive data management tool specifically tailored for the healthcare data industry. This tool aims to address several critical issues in healthcare data management, such as data analysis, organization and security. The objectives include: Designing and implementing a user-friendly interface for loading, inserting and retrieving data, also including patient records, diseases statistics and treatment outcomes. Developing efficient analytics capabilities to extract useful insights from the healthcare data. Ensuring agreement with privacy rules and data security to protect patients confidentiality. Providing a powerful visualization tool to convey healthcare data in a more informative manner.

Report breakdown

The report breakdown summarizes the key sections of the project. It includes:

1. **Problem Statement**: Addressing challenges in healthcare data management, emphasizing the need for a comprehensive solution.
2. **Motivation**: Highlighting the importance of healthcare data in improving patient outcomes and driving advancements in the healthcare industry.
3. **Background**: Exploring the exponential growth of healthcare data and the limitations of existing data management tools.
4. **Objectives**: Outlining the main goals of the project, including designing a user-friendly interface, developing analytics capabilities, ensuring data security, and providing visualization tools.

Related Work

In preparation for our project, we conducted various research and reports on **Canadian Institute of Health Information** about Inpatient Hospitalization, Surgery and Newborn Statistics. This product provides 2022–2023 data on hospitalization rates, average length of stay, high-volume inpatient surgeries and hospitalizations by province/territory, as well as trends on in-hospital births. The utilization of the data include surgery statistics, newborn statistics and

more. Surgery statistics procedure is used to access the high-volume of inpatient surgeries and identify the common surgical procedures like ACLs, Cesarean section delivery and more, performed within the healthcare system. The newborn statistics are monitored to register trends in childbirth and maternal healthcare outcomes in different provinces and territories. By analyzing birth rates and related matters, healthcare providers can assess the information and effectiveness of maternal health procedures and identify areas which can be improved and provide the best possible procedure within the healthcare system.

Before analyzing the raw dataset given by the **Canadian Institute of Health Information**, we must make sure that the data is clear of any missing errors, words are any inconsistencies within the dataset. In this way, we can produce the highest quality implementation and facilitate accurate and effective analysis results. Visitation tools like charts and graphs are utilized to represent the anomalies and findings to give more insights effectively. This **Canadian Institute of Health Information** dataset plays a crucial role in achieving the objectives for our project, in which include understanding the healthcare patterns and trends, identifying areas for healthcare improvements, insights and resource allocations, supporting evidence-based data driven analytics statistics and improving overall and contributing to the development of the healthcare data management tools to improve overall patient outcomes and contribute to a positive outcome in the healthcare industry as a whole.

Thanks to the comprehensive dataset provided by **Canadian Institute of Health Information**, we successfully implemented a robust data management tool capable of filtering and presenting critical healthcare metrics including fiscal year, province, hospitalization rate, and length of stay. This tool serves as a valuable resource for healthcare professionals, data analysts, and the general public, enabling them to access and interpret meaningful insights from the vast repository of healthcare data.

Methodology

Our Approach

In tackling the development of a comprehensive data management tool tailored for the healthcare data industry, our project methodology is firmly rooted in the principles of agile development. This decision is predicated on the agile framework's inherent flexibility, which is crucial for adapting to the evolving requirements and challenges characteristic of developing a healthcare data management application. Agile's iterative process allows for continuous feedback and refinements, ensuring the project remains aligned with its objectives and responsive to stakeholder needs.

Our Assumptions

Our approach is guided by several key assumptions, which have been instrumental in shaping the project's design and implementation strategy:

1. **Data Sensitivity and Privacy Compliance**: The dataset from the Canadian Institute of Health Information (CIHI) is presumed to contain sensitive information. This assumption necessitates a rigorous adherence to privacy laws and ethical considerations in data handling. Our project prioritizes the implementation of robust security measures and privacy-preserving mechanisms to protect individual confidentiality, in strict compliance with legal restrictions against sharing personally identifiable information.
2. **Time Constraints**: We are operating under the assumption that the project timeframe spans just a few months. This limitation emphasizes the need for an agile methodology, allowing for rapid development cycles and the ability to prioritize project deliverables to meet tight deadlines.

Our Environment and Setup

In establishing our development environment, we have carefully selected tools and platforms that enhance collaboration, facilitate efficient coding practices, and enable thorough testing of our healthcare data management tool. Our setup is characterized by the following components:

1. **GitHub CodeSpaces**: Our primary platform for code collaboration is GitHub Spaces. This choice allows our team to efficiently manage our codebase, track changes, and facilitate seamless collaboration across different segments of the project. GitHub Spaces offers an integrated environment for version control and code review, ensuring that all team members are aligned and can contribute effectively, regardless of their geographical location.
2. **OnlineGDB**: For immediate testing and execution of our code. This versatile online compiler and debugger tool supports multiple programming languages, making it an ideal choice for rapid prototyping and debugging. Its accessibility and ease of use allow our team members to quickly test code snippets and modules without the need for extensive local setup, speeding up the development process.
3. **Visual Studio Code**: For local development and more in-depth testing scenarios. This powerful source-code editor supports a wide range of programming languages and frameworks relevant to our project, and it is equipped with features like IntelliSense for code completion and debugging tools. VS Code's versatility and extensive plugin ecosystem enable customized setups that cater to our specific development needs, ensuring that code is efficiently written, tested, and debugged locally when required.

This combination of tools and platforms constitutes a robust development environment that supports our agile methodology, enhances our ability to adapt to project changes, and ensures the high quality and reliability of our healthcare data management tool.

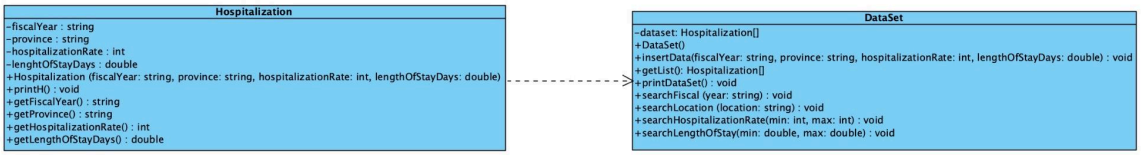
Our Object Oriented Design Approach

In our object-oriented (OO) design approach, we prioritize the integration of principles from various design patterns to enhance the efficiency and resilience of our codebase.

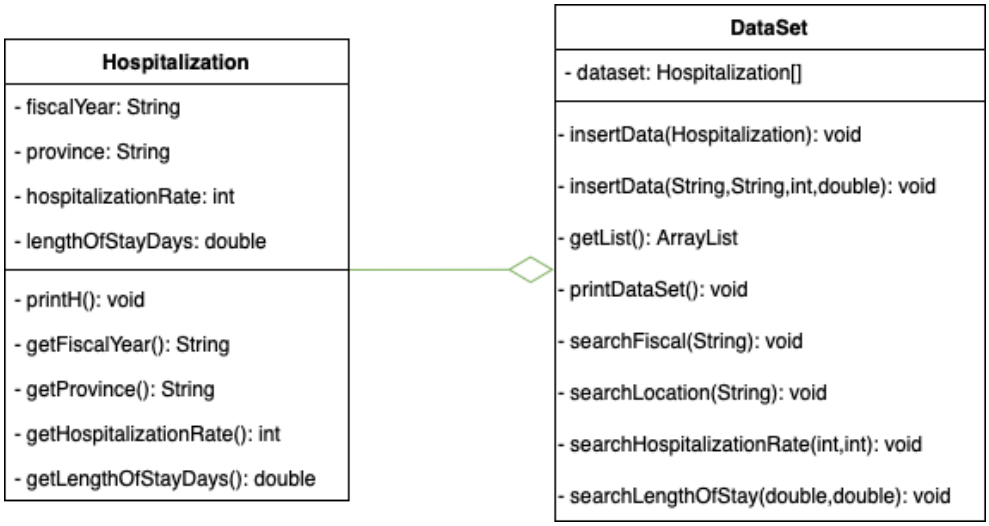
We utilize General Responsibility Assignment Software Patterns (GRASP) to assign responsibilities to classes and objects, ensuring low coupling and high cohesion. Additionally, we incorporate patterns from the Gang of Four (GoF) to further optimize our code. For example, the Singleton pattern helps us manage class instances effectively, while the Factory pattern streamlines object creation logic.

Through the application of these design patterns, we aim to achieve efficiency, resilience, scalability, and maintainability in our software solution, aligning with the objectives of our project effectively.

Examples of OO Design Approach



The Hospitalization class embodies the Information Expert principle by encapsulating essential hospitalization data such as fiscal year, province, hospitalization rate, and length of stay. This design choice consolidates relevant information within a single class, making it the natural choice for handling operations related to hospitalization records. By centralizing data and behavior within the class, it promotes encapsulation, reduces dependencies, and enhances code maintainability. Thus, the class aligns with the Information Expert principle by ensuring that responsibility for managing hospitalization data resides where the most pertinent information is located.



The Hospitalization class is intentionally designed with individual getter methods to encapsulate access to its attributes. This deliberate design choice aims to promote loose coupling between components by abstracting the internal representation of hospitalization data from external entities.

Similarly, the insertData method in the main class encapsulates the logic for inserting data into the dataset. This approach shields external components from internal implementation details, ensuring that changes to the dataset structure or the Hospitalization class do not impact external components.

Prototype

Experimental Setup/Demonstration

The prototype in this case is the healthcare data management system, which we will describe using its components, how it executes, and how it is set up/how the data can be interpreted. We will also discuss how we will use TDD (Test-Driven Development) and other refactoring practices so that the system is easier to use, scale, and become more reliable overall.

Prototype Components

The prototype of the hospitalization data management system has the following components:
Main: Allows for interactions between the user interface and dataset.
DataSet: Manages the hospitalization data, and insertion, retrieval, and searching functionalities.
Hospitalization: Holds hospitalization records and information like fiscal year, province/territory, hospitalization rate etc.

Execution and Experimentation

To execute the prototype, users need to interact with the command-line interface from main. They will choose between options like printing the dataset or specific data searching. During development, we will use TDD practices, writing tests before implementing the production code. By using frameworks like JUnit, the system will also perform more consistently.

```java
// Example of TDD
import static org.junit.Assert.*;
import org.junit.Test;

public class DataSetTest {
    @TDDTest1
    public void testInsertData() {
        DataSet dataset = new DataSet();
        dataset.insertData("2022-2023", "Ontario", 6666, 6.9);
        assertEquals(1, dataset.getList().size());
    }
    @TDDTest2
    // Etc.
}
```

Another practice we will include is refactoring, which continuously improves the code quality without altering the external behavior. By applying small behavioral changes in the code, and then running the tests again after each refactor, the codebase won't have any duplications.

```java
// Example of Refactoring
public class DataSet {
    // Code in DataSet
    // Refactored method: Extract Method
    private void printHeader() {
        System.out.println("\n-----------------------------------------------------------------------------------------------------");
        System.out.println("| Fiscal Year | Province/Territory      | Hospitalization Rate Per 100,000 | Average Stay (Days) |");
        System.out.println("-----------------------------------------------------------------------------------------------------");
    }
    // Now update printDataSet() to use the extracted method
    public void printDataSet() {
        printHeader();
        for (Hospitalization h : this.dataset)
            h.printH();
        System.out.println("-----------------------------------------------------------------------------------------------------\n");
    }
    // Etc. (Other refactoring practices)
}
```

Results and Interpretation

By following TDD and refactoring practices, we will ensure that the DMT system is properly tested. The application's reliable design and how it follows proper coding practices also allows for tweaks and scalability of the system in the future. The below appendix is an example of a successful test run of the system by inputting normal values.

2022-2023	New Brunswick	7,231	8.4
2022-2023	Quebec	6,761	6.9
2022-2023	Ontario	6,666	6.9
2022-2023	Manitoba	7,316	9.9
2022-2023	Saskatchewan	9,590	7.1
2022-2023	Alberta	7,444	7.9
2022-2023	British Columbia	6,908	7.4
2022-2023	Yukon	10,178	5.1
2022-2023	Northwest Territories	12,407	6.8
2022-2023	Nunavut	15,342	3.8
2022-2023	Canada	7,000	7.3
2021-2022	Newfoundland and Labrador	7,328	7.9
2021-2022	Prince Edward Island	7,808	9.2
2021-2022	Nova Scotia	6,796	8.6
2021-2022	New Brunswick	7,297	8.2
2021-2022	Quebec	6,763	6.9
2021-2022	Ontario	6,557	6.8
2021-2022	Manitoba	7,449	9.6
2021-2022	Saskatchewan	9,272	7.0
2021-2022	Alberta	7,527	7.5
2021-2022	British Columbia	6,994	7.2
2021-2022	Yukon	9,834	5.1
2021-2022	Northwest Territories	12,947	7.2
2021-2022	Nunavut	14,188	4.4
2021-2022	Canada	6,983	7.2
2020-2021	Newfoundland and Labrador	7,146	7.7
2020-2021	Prince Edward Island	7,758	7.8
2020-2021	Nova Scotia	6,723	8.6

And in the case that values are out of range, undefined, misspelled or other as outlined in the TDD test code, the appropriate message shall be displayed for each.

Test Case: Searching for data in a non-existing fiscal year
Input: Fiscal year = "2025-2026"
Expected Output: "No data found for fiscal year 2025-2026."
Actual Output: "Error: No data found for fiscal year 2025-2026."

Test Case: Searching for data in an invalid province/territory
Input: Province = "XYZ"
Expected Output: "No data found for province XYZ."
Actual Output: "Error: No data found for province XYZ."

Test Case: Searching for data with an invalid hospitalization rate range
Input: Minimum rate = 20, Maximum rate = 10
Expected Output: "Error: Invalid hospitalization rate range."
Actual Output: "Error: Invalid hospitalization rate range. Please ensure that the minimum rate is less than or equal to the maximum rate."

Test Case: Searching for data with an invalid length of stay range
Input: Minimum days = 5.5, Maximum days = 3.5
Expected Output: "Error: Invalid length of stay range."
Actual Output: "Error: Invalid length of stay range. Please ensure that the minimum day(s) is/are less than or equal to the maximum day(s)."

Discussion

Evolution Through Phases

The development of the dashboard for the **Canadian Institute of Health Information** went through a very natural progression and evolution over the course of its lifetime. As anticipated in any major project, moving from one phase to another created a need for adjustments over the course of its development. Over the course of its development, some functionalities were enhanced while others were altered or removed. Through this process we were able to refine the product, ensuring it stayed in line with our project scope.

Planning versus Reality

Despite the amount of planning that was involved in its development, it is often impossible to fully plan for every possible outcome. This caused us to rethink constantly through its development to make sure the project was achieving our goal.

Project Outcomes

The end result, although a bit different from our initial concept, is proof of our team's ability to adapt and maintain focus on creating a valuable tool for the **Canadian Institute of Health Information**. Throughout its design and development, continuous testing was implemented to verify functionality and the overall usefulness of the product. The results from our continuous testing helped outline and identify the needs of our users.

Reflection

The overarching journey from project start to final product was refined by our adaptation and refinement. The challenges we encountered along the way were invaluable to its development.

Conclusion

Our core goals in the beginning were to develop a database management system where credible users could insert data into a database, view said data with a search functionality, switch between different unique datasets, as well as having a way to display different graphs to better illustrate the meanings of the data. Over the project, our goals shifted as we had difficulty figuring out where to start. What we were able to achieve was a system that has a dataset that can be easily manipulated by an admin, and displayed neatly to a user. That user can not only display the dataset, but also search for any specific attribute of the dataset to obtain specific pieces of that data. We are happy with what we achieved but there is much that has not yet been added that was cut short during the development process.

In its *current* state, there is only one dataset that is implemented in the program. This is one key feature that is of high importance that we failed to reach in time. This kind of system is complicated, and time consuming to implement. For the best result, we would have wanted to add an abstract element to the core dataset class to allow for a list of data that is not bound by one specific dataset's properties, but rather able to change to fit any kind of dataset with varying amounts and types of attributes. This would take a long time to develop and was ultimately pushed aside to allow us to focus on getting the simpler program fully functional. In addition to this, we also were not able to add graphs or charts to the program to display data in a visually pleasing manner. This was due to the fact that we were working with a simple terminal to display output, and with the current software it would look very ugly to add. To be able to add new graphs to display data, we would first need to shift our development elsewhere where we could implement a proper user interface with a far cleaner display. This UI could then let us use different tools to add the graphs we want. The number one reason that this was skipped was simply that none of us were familiar with any software used to implement a UI.

Finally, a user login was not added for simplicity as we felt it was not of high priority and that this project could be fine without it for the time being. Users would in the end be able to access and view several datasets but would not be able to add any of their own. After looking back at what we achieved with this project, we are content. Although there are features that remain to be seen, overall, the product in its current state works well and maintains core programming principles. We strongly believe that our product once finished could be of significant use for having readily available data pertaining to health information for anybody to view for free.

Future Direction

Regarding future project plans, the next iteration would entail new key features that we were not yet able to achieve. In this new iteration, the primary focus would be on implementing the ability to switch between different datasets. We anticipate that this will not be a straightforward process but will be doable. This feature is important so that we can expand the amount of data that a user can view through our program. Once this is implemented properly, we would then move the program into a new system that can use a nice, organized user interface. This implementation would allow for a far superior user viewing experience and would allow us to further expand the project through the addition of graphs that better highlight the data. That would be all regarding the next iteration, it is ambitious, but would be required to speed up the development process.

In addition to this, we believe that our program could become a successful commercial product. Although there are many websites that display data, we can use our program to target new and unused data that could differentiate us from any competitors. This way, our program would stand out as we could easily add increased data to it every day. With this in mind, we believe our program has a place in the market.

References

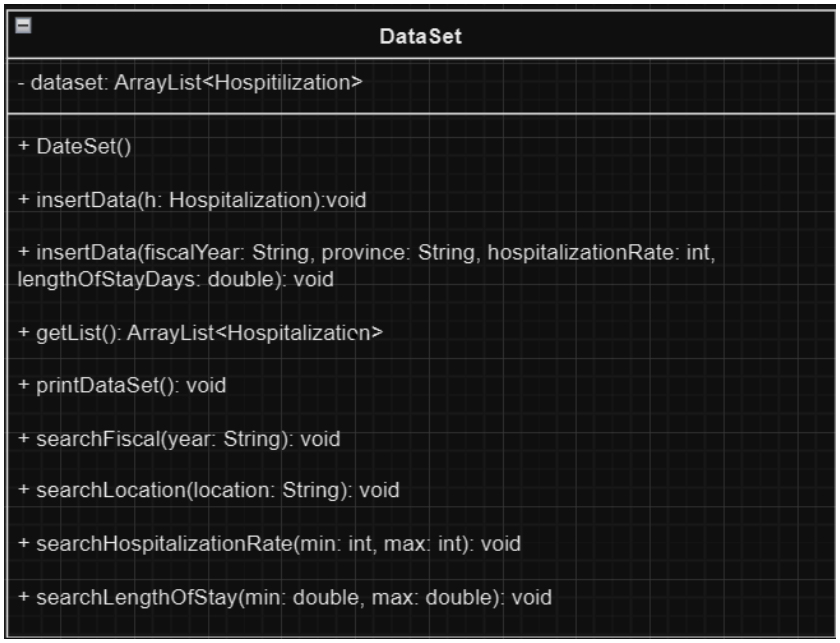
[1] Canadian Institute for Health Information, “CIHI,” *Cihi.ca*, 2017. https://www.cihi.ca/en

Appendices

Appendix A - Group Work

We used a variety of methods and tools for this project. We held regular meetings every Saturday at 1 pm on Discord to talk about project progress, to assign tasks to each member then near the end we discussed any issues or concerns we had. The tool we used to track progress is Google Docs. Google Docs has features where we can see how much work each person does and can see what other members are doing.We used agile methodology when making this project, this made us able to change things and improve them way faster. Discord was important in our project because it allowed open communication and feedback for our project. We used Git and Github to have a personal repository containing the code for the project. For testing we commonly used an online Java compiler to run and develop the code.

Appendix B - Methodology



The DataSet class exemplifies high cohesion, with its methods tightly focused on managing hospitalization data effectively. Data management methods, including the constructor and insertData operations, handle the addition of new hospitalization records to the dataset. Retrieval and printing methods, such as getList and printDataSet, respectively, provide access to and display the dataset contents. Additionally, search methods like searchFiscal,

searchLocation, searchHospitalizationRate, and searchLengthOfStay facilitate dataset queries based on specific criteria. This cohesive design streamlines the class's functionality, promoting clarity, maintainability, and reusability by organizing related operations within a single unit.

Appendix C - Special Acknowledgements

We acknowledge the usage of the following academic licenses, the first one goes to OnlineGDB. This software allowed us to easily have access to Java without any set up. This tool also allows multiple files with an option of a text file, this goes above and beyond any other online compiler. Nothing but positive experiences using this tool, our project would have taken longer in the coding development stage. This software made the code easy to share and run which means there are more cycles between feedback.