NATIONAL TEACHERS COLLEGE

Graduate School – School of Teacher Education
629 Nepomuceno St. Quiapo, Manila1001 Metro Manila

ECOTRACKER:
SGD 12: Responsible Consumption and Production

A Final Project

presented to the faculty of the

School of Arts, Science and Technology

National Teachers College

Quiapo, Manila

In Partial Fulfillment of the Requirements

in Information Management

for the Degree of

Bachelor of Science in Information Technology

Submitted by:

Efe, Aldrich Allister R.

Submitted to:

Ms. Justin Louise R. Neypes

December 2025

I. INTRODUCTION

Project Overview & UN SDG Target

EcoTracker is a data-focused waste management system aimed at assisting users in organizing, analyzing, and overseeing various forms of waste. The application retrieves waste records from a CSV file, organizes them with different data structures (linked lists, queues, trees), and offers helpful features like sorting, searching, category management, and data visualization. Users can add new waste entries, manage categories, and export updated datasets for later use.

The project showcases real-world uses of essential Data Structures and Algorithms (DSA), such as linked lists for data storage, queues for managing collections, binary search trees for organizing categories, and sorting/searching algorithms for processing data. Through the simulation of actual waste tracking, the system aids users in comprehending how structured data can be organized effectively for making decisions.

This initiative backs SDG Target 12 by offering a platform that organizes and evaluates waste data, assisting users in understanding waste generation trends and classifying materials for appropriate disposal or recycling. Through enhancing awareness and offering organized data management, EcoTracker aids in minimizing environmental effects and encouraging sustainable waste practices.

Problem Statement

The EcoTracker initiative addresses multiple actual waste management issues by assisting users in organizing, analyzing, and monitoring various waste types. Numerous communities and organizations face difficulties in effectively managing recyclable, hazardous, and organic waste due to data frequently being disorganized or untracked entirely. This results in ineffective waste disposal, overflowing waste management sites, and heightened environmental contamination.

The system enhances decision-making by enabling users to upload, sort, search, and classify waste data. It aids in determining which materials are most common, which categories need immediate disposal, and how

waste patterns evolve over time. Such organized data is crucial for designing recycling initiatives, minimizing waste generation, and stopping hazardous substances from polluting the environment.

II. REQUIREMENTS & ANALYSIS

Functional Requirements and Non-Functional Requirements

ID	Feature	Alignment
FR 1	The system is able to load waste records from an external CSV file at program start up.	Prelim DSA
FR 2	The system allows the users to perform processing operations such as sorting waste data, searching for records, and maintaining waste categories.	Finals DSA
FR 3	The system implements and demonstrates at least three required Data Structures and Algorithms (DSA) concepts.	Core DSA
FR 4	The system displays all results in a clear, formatted output on the console and support saving updated records to an external file.	User Interface

ID	Feature
NFR 1	**Usability:** The system provides a clear, menu-driven console interface that is easy to navigate for users with basic computer knowledge.
NFR 2	**Reliability:** The system handles invalid file paths, malformed CSV entries, and incorrect user inputs without crashing.
NFR 3	**Maintainability:** The system's codebase shall be modular, with separate .cpp and .h files for each component.

Data Requirements

The EcoTracker program requires a structured CSV file as its primary input, containing waste-management records that the system processes upon startup. Each line in the file must follow a consistent format with five comma-separated fields: id, type, weight, date, and priority. These fields represent a waste item's unique identifier, category, mass in kilograms, recorded date, and assigned priority level. The program reads each line, validates the values, and stores the records internally using linked lists, arrays, queues, and trees to support the system's various data-processing features.

To ensure efficient performance and prevent data overflow, the system is designed to handle a maximum of 50 waste entries. If more than 50 records are present in the CSV file, only the first 50 are loaded. All input files must follow the proper CSV structure and be placed in a recognized folder, such as

INPUT_DATA/data.csv. This structured and size-controlled approach ensures that EcoTracker operates reliably while demonstrating the required data structures and algorithms for the project.

Complexity Analysis

The EcoTracker program uses several core data structures: inked lists, arrays, queues, and binary search trees. Each operation has an expected performance behavior that can be represented using Big O notation. Loading the CSV file into a linked list takes O(n) time because the system reads each line once and inserts the record at the head of the list. Since the program caps the dataset at 50 records, this operation always remains efficient.

Sorting operations, such as those performed in the Waste Sort module, rely on converting the linked list into a temporary array and applying a comparison-based sorting algorithm. Searching depends on the chosen structure: linear search in the linked list runs in O(n) time, while tree-based operations for category lookup take O(log n) time on average due to the hierarchical nature of binary search trees.

III. Design Specification

Core Data Structures Used

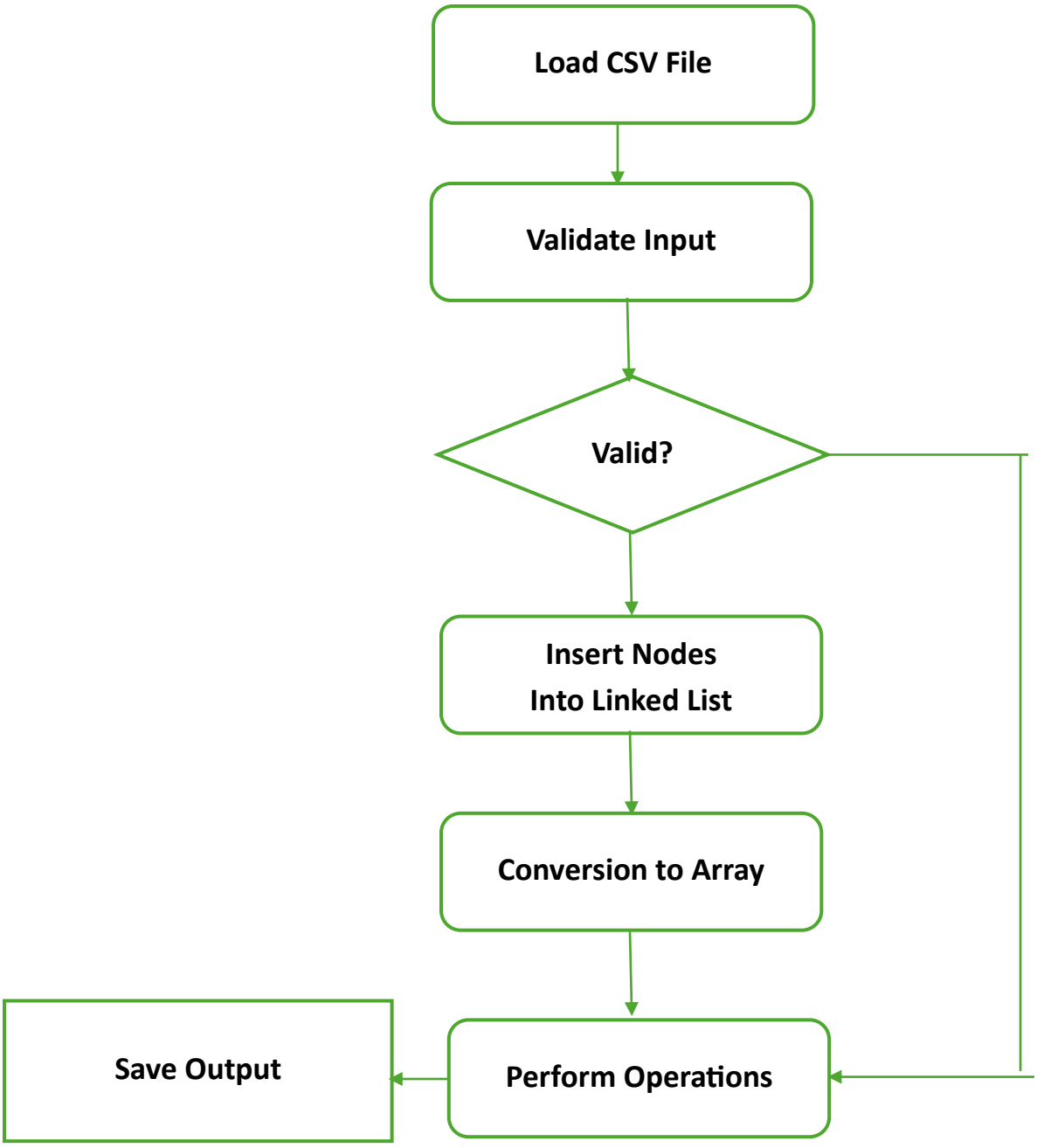
The data structures and algorithms used in EcoTracker were selected because they match the real-world behaviors the system needs to perform while keeping the program efficient, organized, and easy to maintain.

A linked list was chosen for storing waste items because it allows fast insertion of new records from the CSV file and from user input. Since EcoTracker does not require random access but often adds items at the beginning, a linked list is more efficient and simpler than an array for this task.

An array is used when sorting is needed because arrays work extremely well with efficient sorting algorithms, Linked lists cannot be sorted efficiently without converting them to arrays first, so using arrays here gives the best performance and simplicity.

A queue is used for the waste collection scheduling feature because it naturally models a real world "first-come, first-served" process. Items enter the queue in order and are processed exactly the same way real collection tasks are handled. Meanwhile, a binary search tree is used for organizing waste categories alphabetically because BST operations such as insertions, searches, and traversals are efficient at O(log n) average time and naturally produce sorted output.

Algorithm Flowchart



Module Breakdown

EcoTracker program is built around several custom C++ classes that work together to manage, organize, and process waste-management data. At the core is the **WasteItem** class, which represents a single waste record and is used by every other component. These items are stored dynamically in **WasteList**, a singly linked list that serves as the system's main data storage, supporting file loading, saving, insertion, and conversion to arrays. When sorting is needed, the **WasteSort** class receives an array version of the list and organizes the items based on ID, weight, or priority using efficient sorting algorithms. Searching operations are handled by the **WasteSearch** module, which retrieves specific records without altering the stored data. Meanwhile, the **CollectionQueue** class models real-world scheduling through a queue that follows first-come, first-served processing of selected waste items. Lastly, category names are organized alphabetically in the **WasteCategoryTree**, a binary search tree that supports quick insertion and traversal. Together, these classes create a modular, structured system where each component performs a distinct role, allowing the EcoTracker program to handle data efficiently and cleanly.

Conclusion

The EcoTracker system effectively illustrates how essential data structures and algorithms can be utilized to address practical issues in waste management and sustainability. Through the use of linked lists, sorting algorithms, queues, and binary search trees, the program effectively manages functions like data loading, waste record organization, collection scheduling, and material categorization. These characteristics align with the objectives of UN SDG 12 by fostering responsible consumption, enhancing waste monitoring, and allowing for improved decision-making in environmental management. In general, EcoTracker emphasizes the significance of applying computational tools to tackle environmental challenges and demonstrates how well-crafted algorithms can enhance the effectiveness, scalability, and accessibility of data-driven solutions.