1. Object Tracker Implementation Documentation

The Object Tracker is a command-line application that demonstrates a linked list implementation for storing, searching and sorting objects with name, ID and date attributes. It provides an interactive interface for users to search objects by name and sort them by date

2. Prerequisites

This section outlines the requirements needed to run the CLI application. This includes library imports and packages needed

Installation Requirements

- Python 3.7 or higher

Required Packages

1. Faker
    a. using pip command in terminal: `pip install faker`

Imports

1. Uuid
    - Provides the ability to generate unique identifiers for each object
2. sys
    - System-level operations
3. typing.optional
    - To indicate attribute that can either contain a node or none
4. typing.list
    - To specify return types that return lists
5. dataclass
    - Provides decorator
6. faker
    - Generate realistic fake data for testing or implementation

2. Data Structure

Node Class

The Node class is the building block of the linked list. It uses the @dataclass decorator. It has the following variable declarations:

- **data**
    - A dictionary containing the object's information (id, name and date)
- **next**
    - A reference to the next node in the list, or None if it's the last node in the Linked list

LinkedList class

The LinkedList class manages a collection of Node objects, providing operations to manipulate and access the data.

It has the following methods:

- **__init__(self)**
  - For initialising an empty linked list with a None head and size of 0
- **append(self, data: dict)**
  - Adds a new node containing the provided data to the end of the linked list
- **search_by_name(self, name: str)**
  - Searches the linked list for objects with matching names and returns a list of matching data dictionaries
- **to_list(self)**
  - Converting of linked list to a python list
- **from_list(self, data_list: List[dict])**
  - Rebuilding the linked list from a list of data
- **sort_by_date(self, ascending: bool = True)**
  - Sorts the linked list by date using python inbuilt sort from a list of data and returns the sorted list

3. Utility Functions

Utility functions are used to assist in formatting, displaying and generation of data

- **generate_random_objects()**
  - Creates a linked list populated with random objects using Faker Library
- **format_date(dt)**
  - Formats a datetime object for display
- **display_objects(objects)**
  - Display a list of objects
- **print_menu()**
  - Display the main menu options

4. Usage of application

The function main() runs the interactive CLI program.

The program flows are as described

1. Initialises a linked list with random data in a dictionary (id, name and date)
2. Display all objects
3. Presents a menu for user interaction to select 1-3:
   a. Search Object by name
   b. Sort Objects by date
   c. Exit the program
4. Processes user input and performs the requested operations
5. Loops back to the while loop menu in 3

To run the program, in a command prompt or cli, type `python object_tracker.py`