HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

### FACULTY OF INTERNATIONAL EDUCATION

### 

**FINAL PROJECT**

**PHONE MANAGEMENT SYSTEM**

Subject: **DATA STRUCTURES AND ALGORITHMS**

**LÊ VĂN NHÂN**

**Student ID**: **23AD11044**

**Major**: **INFORMATION TECHNOLOGY**

**Supervisor**: **TRẦN QUANG KHẢI**

Ho Chi Minh City, 5 December 2024

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

### FACULTY OF INTERNATIONAL EDUCATION

### 

**FINAL PROJECT**

**PHONE MANAGEMENT SYSTEM**

Subject: **DATA STRUCTURES AND ALGORITHMS**

**LÊ VĂN NHÂN**

**Student ID**: **23AD11044**

**Major**: **INFORMATION TECHNOLOGY**

**Supervisor**: **TRẦN QUANG KHẢI**

Ho Chi Minh City, *5th* December 2024

Lecturer’s comments

**................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ ................................................................................................................................................**

**................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................ ................................................................................................................................................**

|  |  |
| --- | --- |
|  | THE SOCIALIST REPUBLIC OF VIETNAM  **Independence – Freedom– Happiness**  -------- |

*Ho Chi Minh city, 5th December, 2024*

**ACKNOWLEDGEMENTS**

The author of this report would like to express deep gratitude to Mr. Trần Quang Khải for his inspiration, guidance, and valuable feedback, which have played a crucial role in the successful completion of this project. His insights, support, and dedication in offering essential knowledge and skills during the study have greatly contributed to the achievement of this work.

A sincere thank you is also extended to family and friends for their constant encouragement, support, and care throughout the entire journey.

The author also wishes to express heartfelt thanks to everyone who has provided assistance and support during this process.

*Ho Chi Minh city, 5th December 2024*

REPORT AUTHOR

Mục lục

Mục lục hình ảnh

**Acronym**

|  |  |
| --- | --- |
| **Binary Search Tree** | **BST** |
|  |  |

**Chapter 1: INTRODUCTION**

* 1. **Introduction**

Mobile phones have become an essential part of modern life, offering convenience and constant connectivity to users. From basic devices primarily designed for calling and texting, mobile phones have evolved into advanced smartphones with features such as touchscreens, high-quality cameras, powerful processing capabilities, and fast internet connectivity. Today, mobile phones are not only communication tools but also multifunctional devices that assist in entertainment, work, and daily tasks. The rapid development of technology, combined with increasing consumer demand, continues to drive the mobile phone industry to innovate and improve, delivering smarter and more versatile products.

* 1. **Problem Statement**

The primary challenge of this project is to effectively manipulate a dataset of phone information using a Binary Search Tree (BST). The project aims to implement various operations such as inserting, updating, and deleting phone entries, along with searching, sorting, and filtering data based on different criteria. The main problem is how to design a BST to support these operations efficiently while ensuring data integrity and proper management.

**1.3 Research Objectives**

The goals of this project are:  
• To design and develop a singly linked list data structure for storing and managing phone information.  
• To implement CRUD operations (Create, Read, Update, Delete) on the dataset.  
• To apply approximately four sorting algorithms learned during the course to organize the data.  
• To create functions for searching, filtering, and identifying the largest or smallest values based on specific criteria.  
• To explore the potential for data visualization and generating charts for better data representation.

**Chapter 2: METHODOLOGY**

**2.1 Dataset Overview**

The dataset used for this project is a phone information's dataset that contains various information from 2017 to 2024. The dataset is stored in CSV format, with each entry representing a detail of phone. Data is taken from [this website](https://www.kaggle.com/datasets)

Each anime record includes the following attributes:

• ID: A unique identifier for each phone.

• Brand: A phone company name

• Model: The type of phone (e.g., Iphone16, Samsung galaxy S21).

• Chipset: A device's Central Processing Unit.

• GPU: A device's Graphics Processing Unit

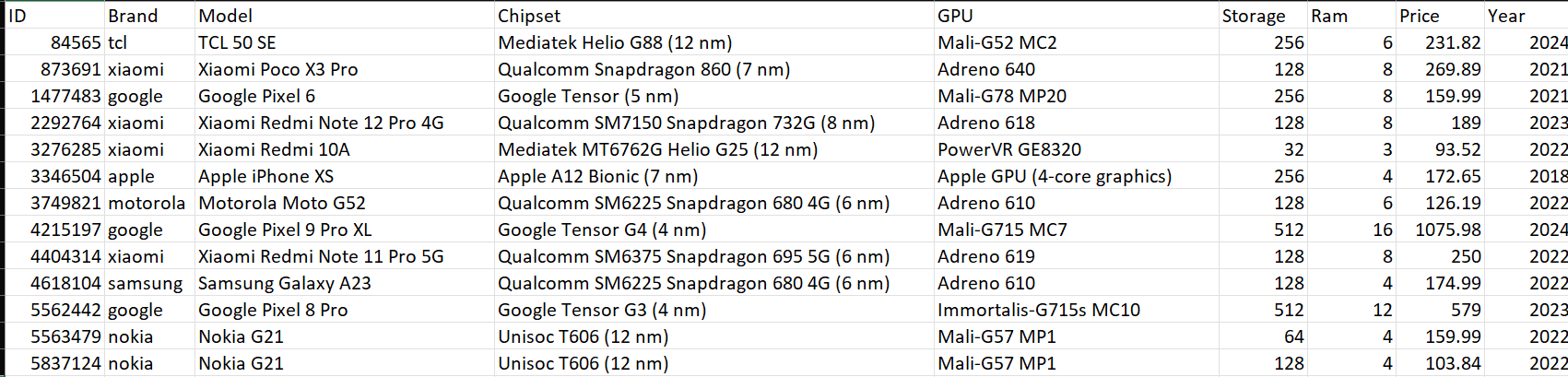
• Storage: Where data is stored

• Ram: A type of volatile memory that allows random read-write access to any location in memory based on a memory address

* Price: The cost of the phone at the time of release

• Year: The year the phone model was manufactured.

An example entry in the dataset is:



**Fig. 2.0 Example of phones**

This dataset provides rich information about different phones, which will be used to implement various operations such as sorting, searching, and filtering. The data was sourced from a public phone database and was pre-processed to remove any irrelevant or missing values before use.

2.2 Binary Search Tree Implementation

**Several benefits**

Using a BST create a “Phone Tree” offers several key benefits:

* **Fast Searching**: BST allows for quick search of phone numbers in the contact list. Since the tree is sorted, searching for a specific number has a time complexity of O (log n), which saves time compared to sequential searching in an unsorted list.
* **Efficient Insertion and Deletion**: Adding or removing a phone number from the BST is straightforward and efficient, with a time complexity of O (log n) in the case of a balanced tree. This allows for easy updates to the contact list.
* **Automatic Sorting**: The BST automatically maintains the order of phone numbers (e.g., by name or phone number), making it easy to list the entire contact list in an ordered fashion without needing to sort it manually.
* **Quick Access to Extremes**: Finding the largest or smallest phone number in the contact list is simple, as it involves just navigating to the far-left or far-right node of the BST.
* **Scalability**: As the contact list grows, the BST maintains stable search and update performance as long as the tree remains balanced. This ensures the system can scale without performance degradation.
* **Efficient Management of Calls and Messages**: The BST can easily integrate with features like storing call history, messages, or additional information related to each contact, making it flexible and efficient in managing data retrieval and updates.

**BST Structure**

Each node in the singly linked list represents a single anime entry and contains two main components:

* Data: This stores the information about the phone, such as: ID. Brand, Model, Chipset, GPU, Storage, Ram, Price and year
* Left – Tree: A pointer that links to the left – tree of root
* Right – Tree: A pointer that links to the left – tree of root

**BST Operations**

Several key operations were implemented on BST to manage the phone data:

* Create: New phone records are added to the BST by inserting a new node at the appropriate position.
* Delete: Phone records can be deleted by searching for the phone by ID and removing the corresponding node from the BST.
* Update: Phone records can be updated by searching for an anime by its ID, then modifying the desired fields (e.g., updating the model, price, ram…).

**2.3. Additional Features**

In addition to the basic CRUD operations, several additional features were implemented to enhance the program's functionality

**Sorting Algorithms**

| **Algorithm** | **Time Complexity (Big-O)** | **Space Complexity** | **Advantages** | **Disadvantages** |
| --- | --- | --- | --- | --- |
| Insertion Sort | - Best: O(n)  - Average: O(n²)  - Worst: O(n²) | O(1) | - Simple and easy to implement.  - Performs well for small or nearly sorted arrays. | - Inefficient for large datasets due to O(n²) complexity. |
| Bubble Sort | - Best: O(n)  - Average: O(n²)  - Worst: O(n²) | O(1) | - Very simple to understand and implement. | - Extremely slow for large arrays.  - Rarely used in practical applications. |
| Merge Sort | - Best: O(n log n)  - Average: O(n log n)  - Worst: O(n log n) | O(n) | - Consistent O(n log n) performance.  - Excellent for large datasets and stable sorting. | - Requires additional memory (O(n)) for merging.  - More complex to implement than basic sorts. |
| Quick Sort | - Best: O(n log n)  - Average: O(n log n)  - Worst: O(n²) | O(log n) (in-place) | - Fastest for most cases, especially large datasets.  - In-place sorting requires less memory. | - Worst-case O(n²) occurs with poor pivot selection.  - Not stable (original order may change). |

Use Merge Sort to sort Models from a --> z alphabetically and Insertion Sort to sort in reverse order. Quick Sort will sort prices ascending and Bubble Sort will sort prices descending

**Searching and Filtering**

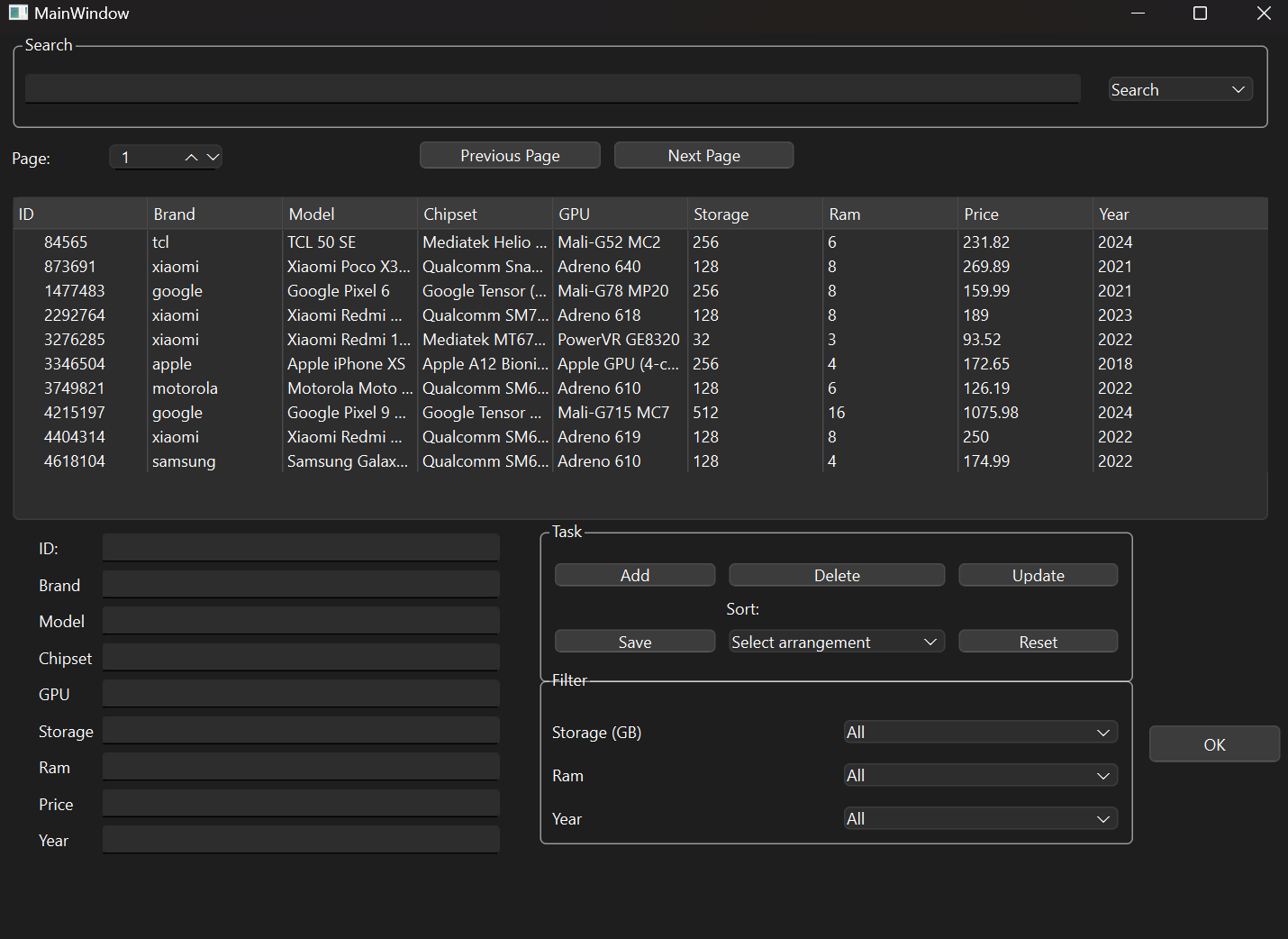
A search feature was introduced to help users locate phones based on specific criteria, such as ID and model. The filtering option allowed users to refine the phone list using multiple factors, including storage, RAM, and year. These functions assist users in quickly finding the phone they are looking for.

**Visualization**

[Images to visualize the data of the BST](Hinh/graphviz.png)

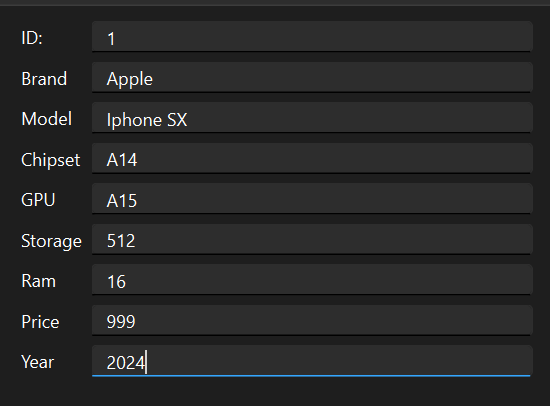
**Chapter 3: IMPLEMENTATION**

**3.1 Interface Overview**

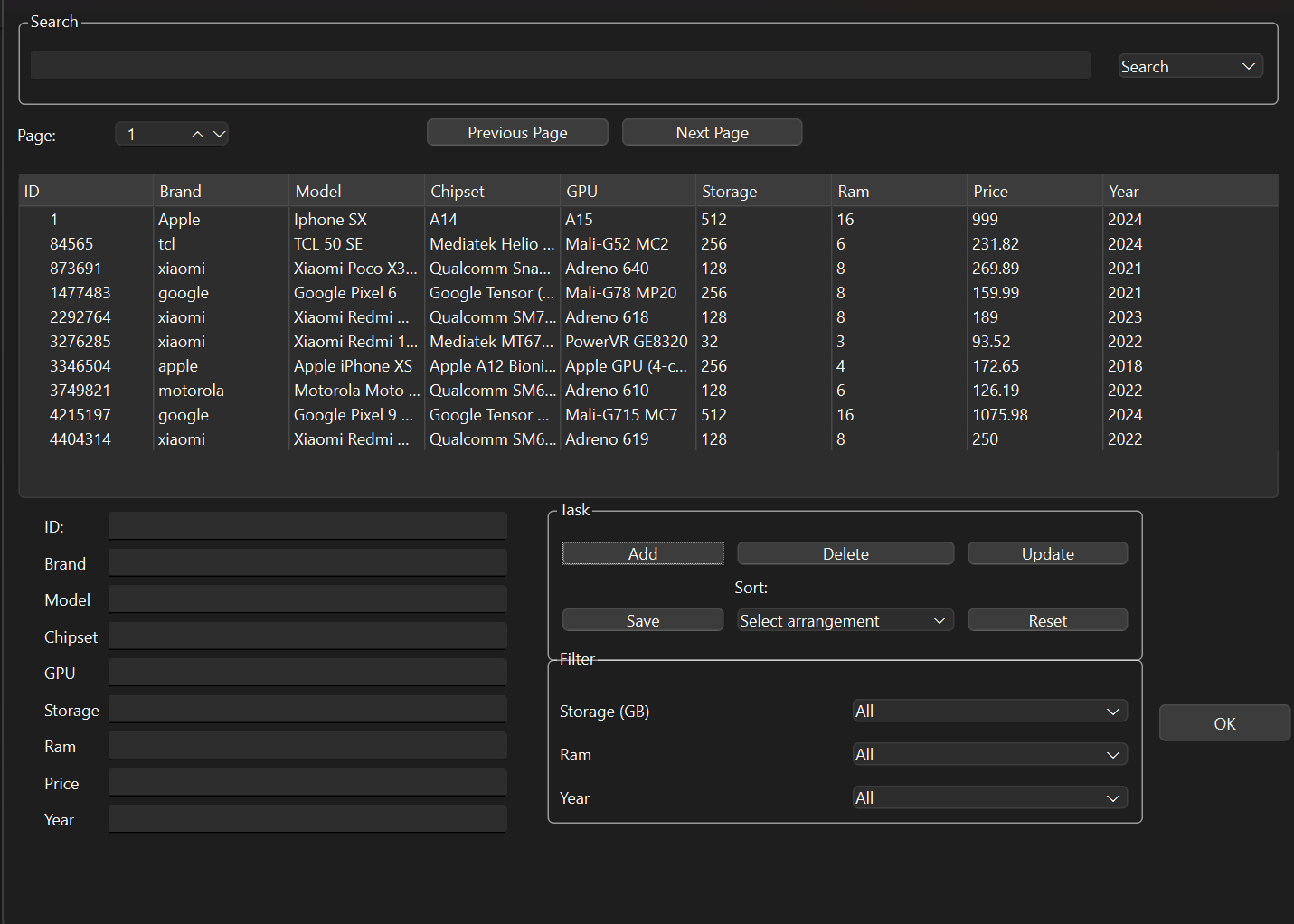
****

**Fig. 3.1.0 User Interface**

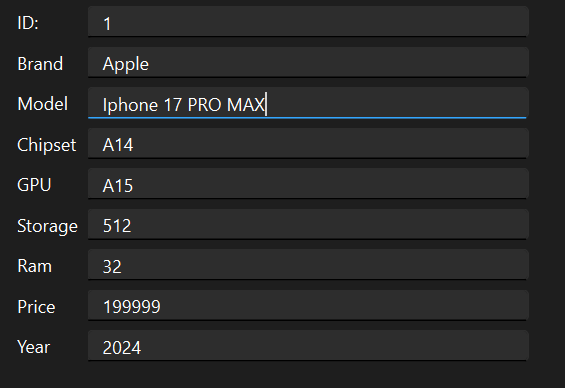
**3.2 CRUD**

****

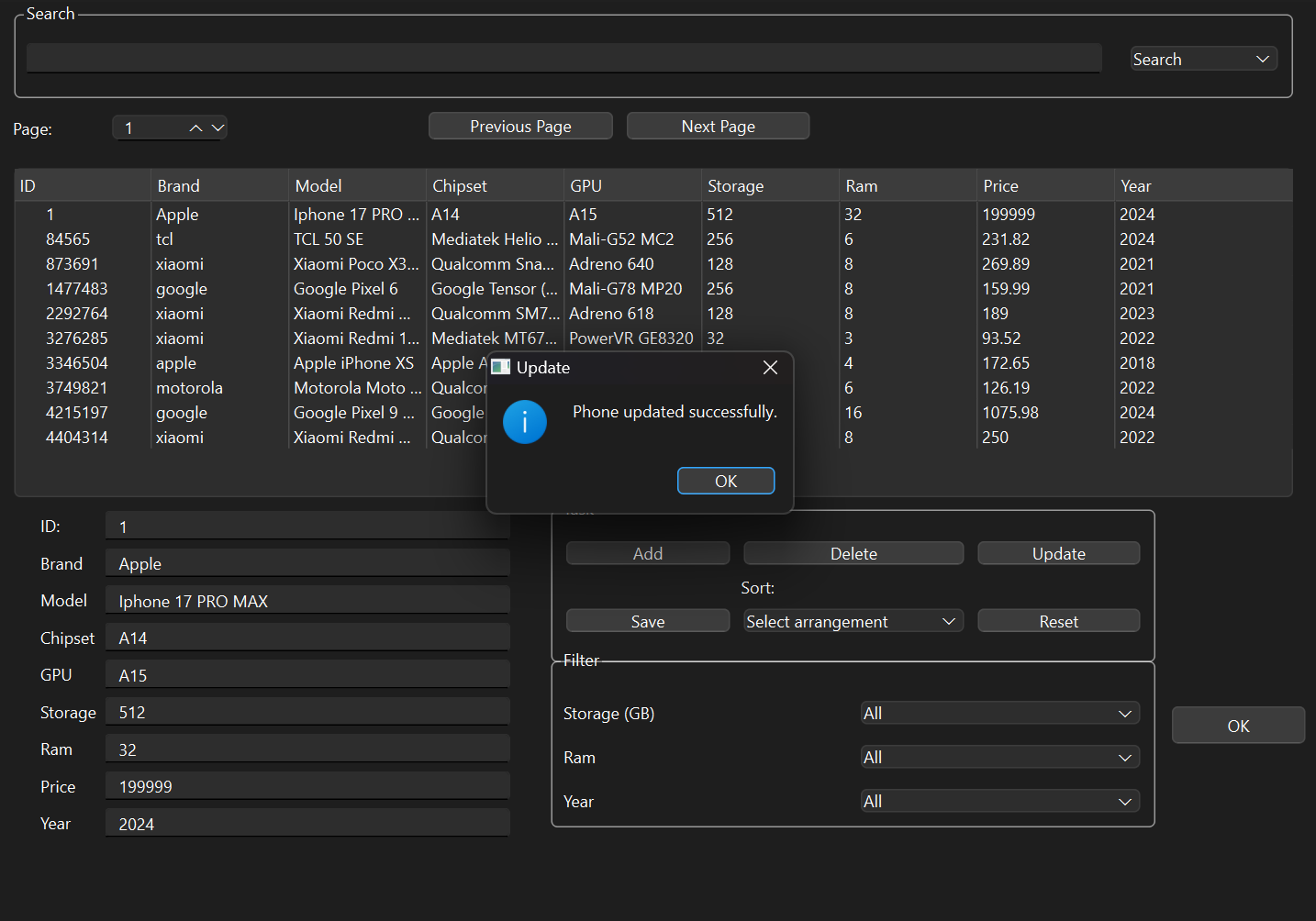
**Fig. 3.2.0 Information on new phone**

****

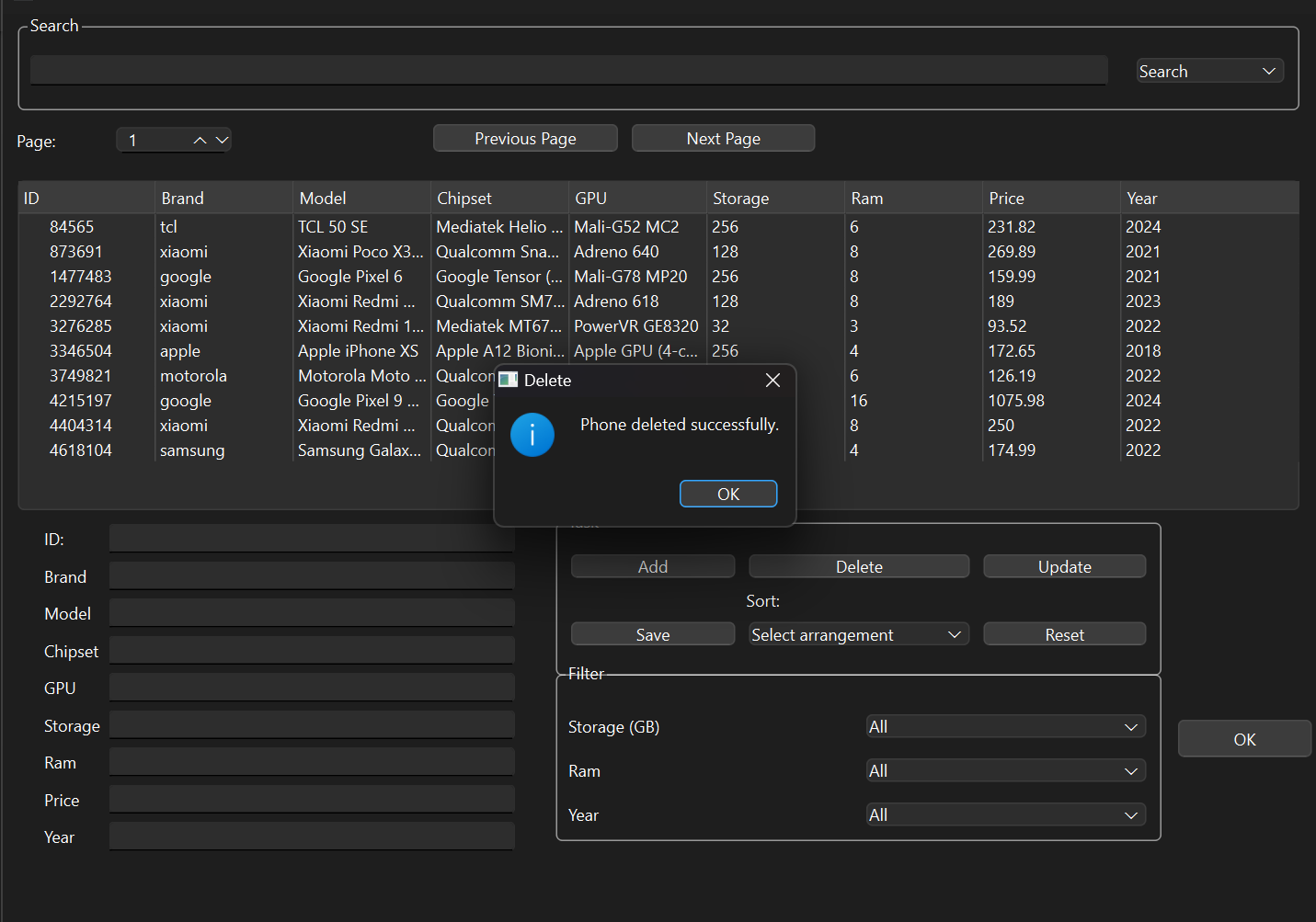
**Fig. 3.2.1 A new phone is added**

****

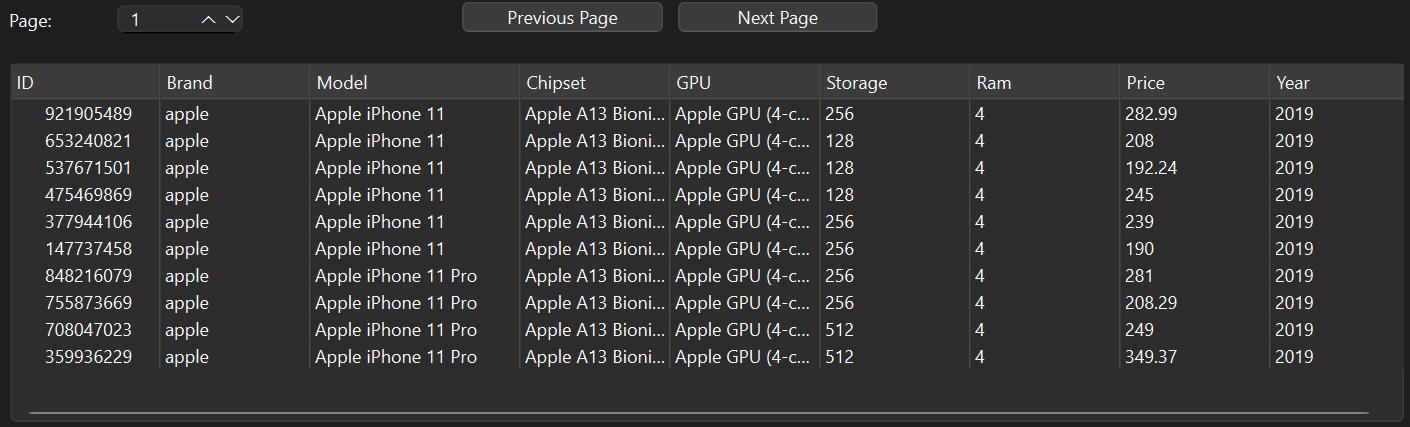
**Fig. 3.2.2 Edit information on the phone**

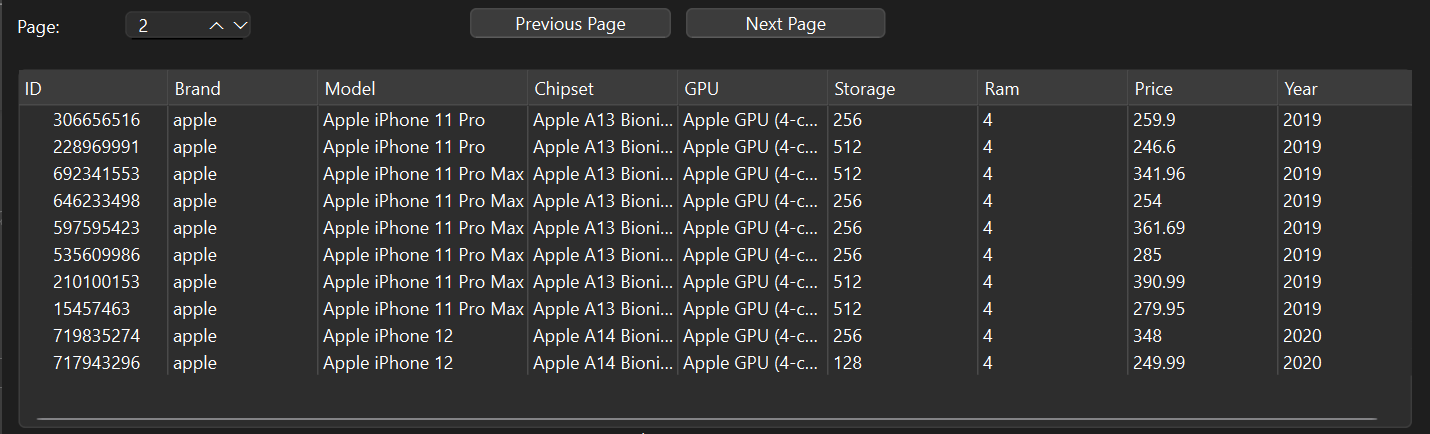
****

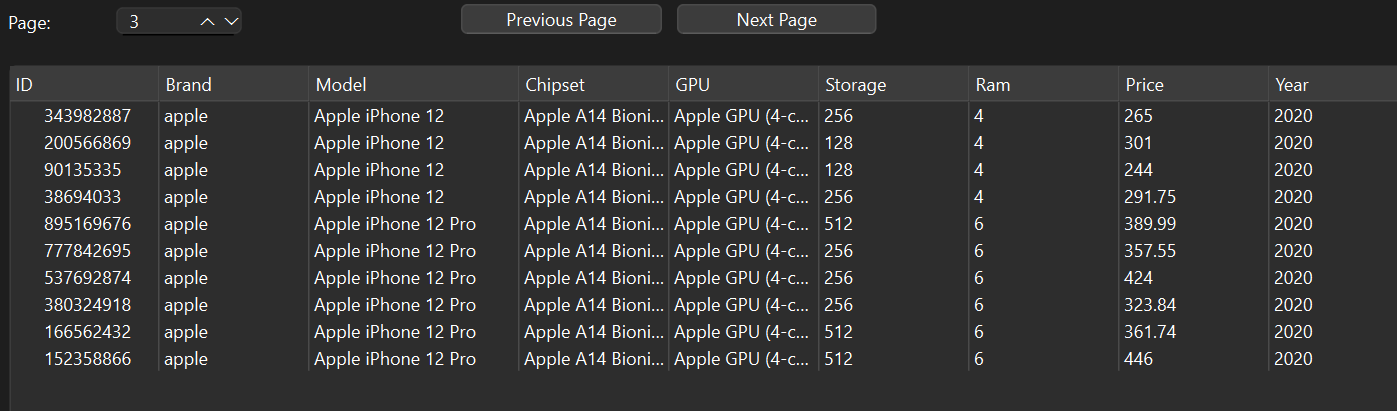
**Fig. 3.2.3 Edit phone successfully**

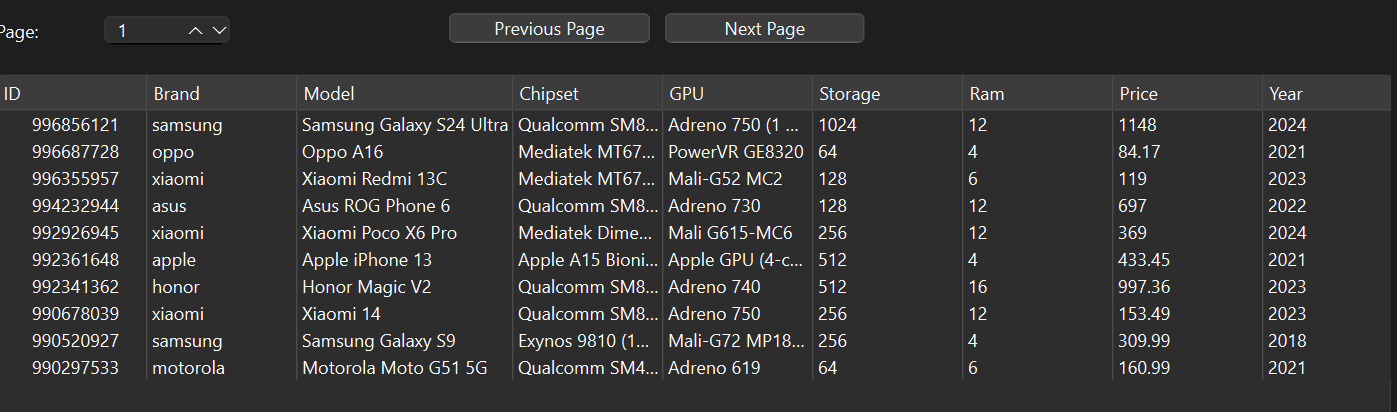


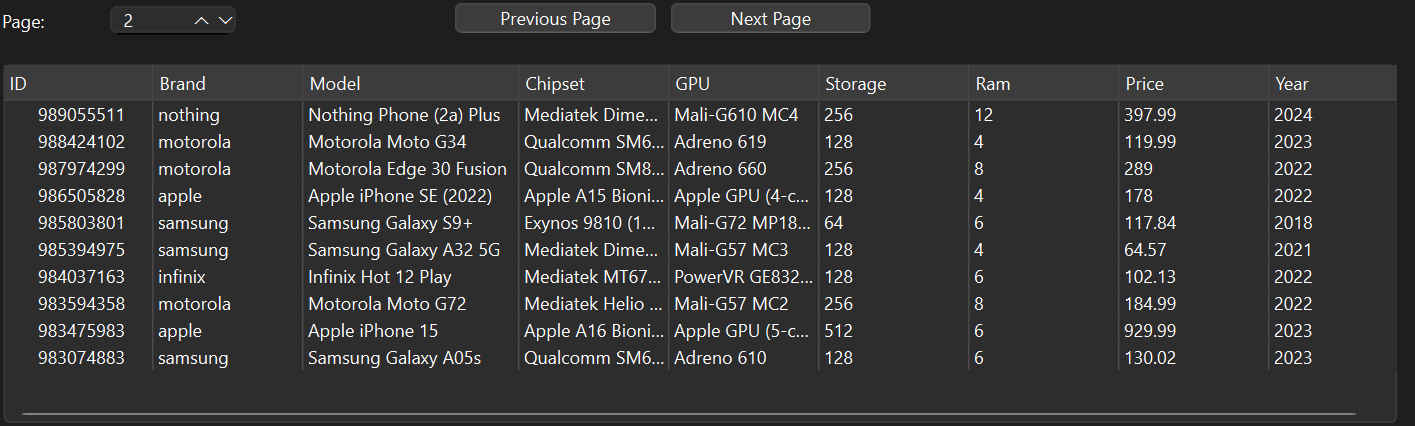
**Fig. 3.2.4 Remove the phone successfully**

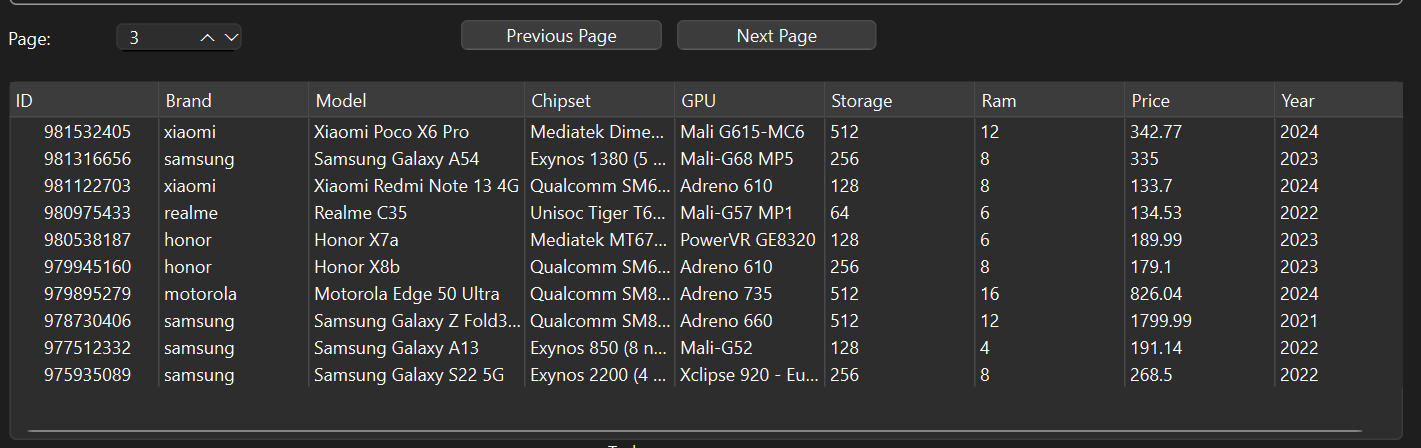
****

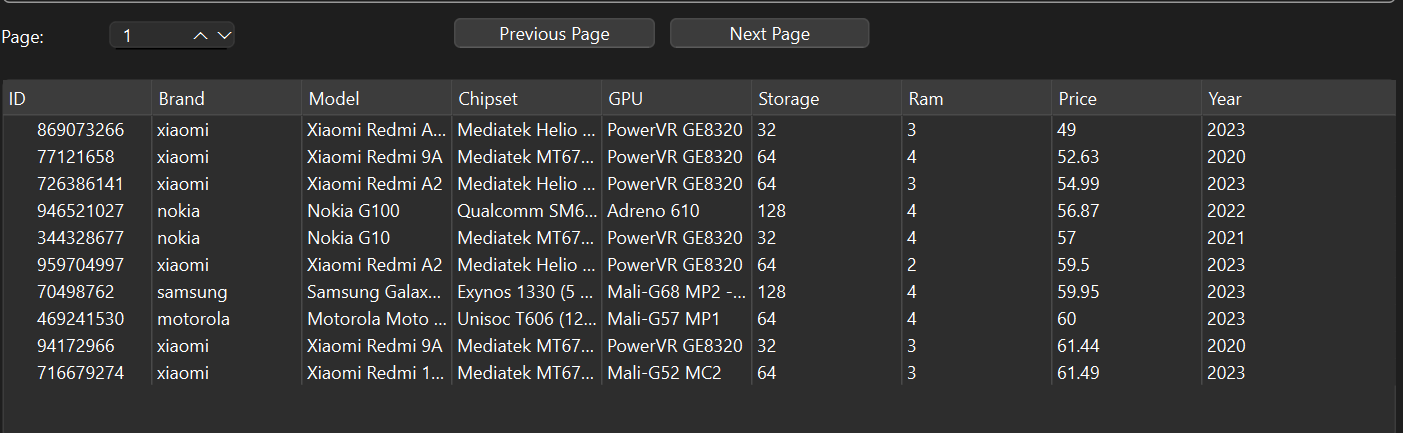
****

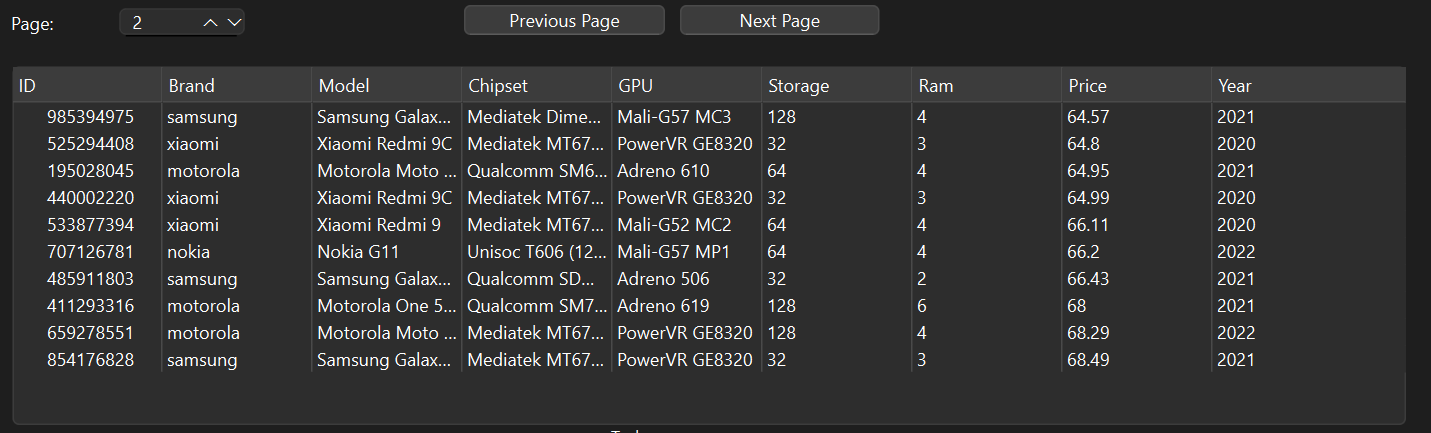
****

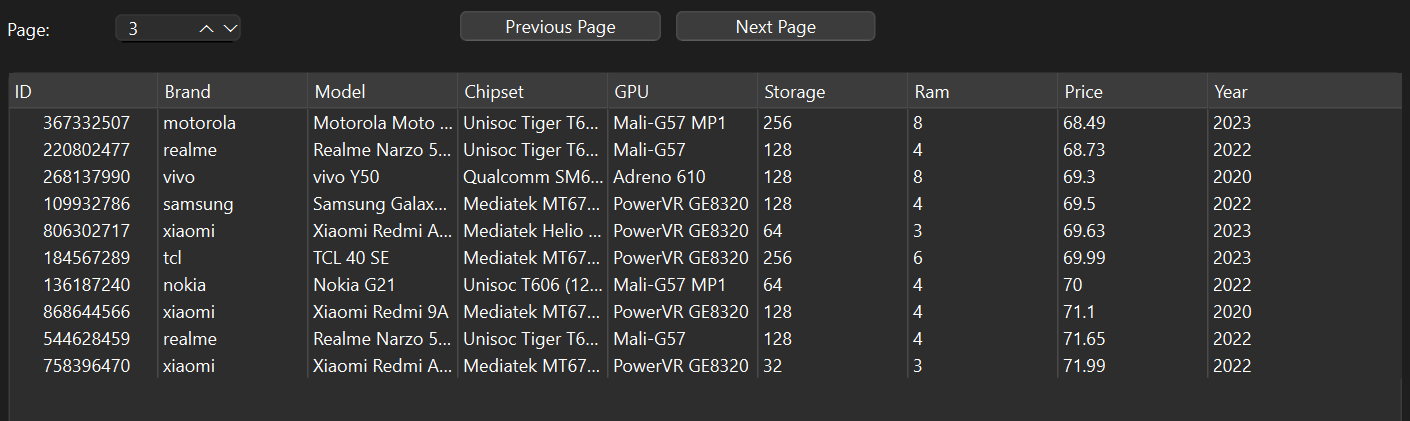
****

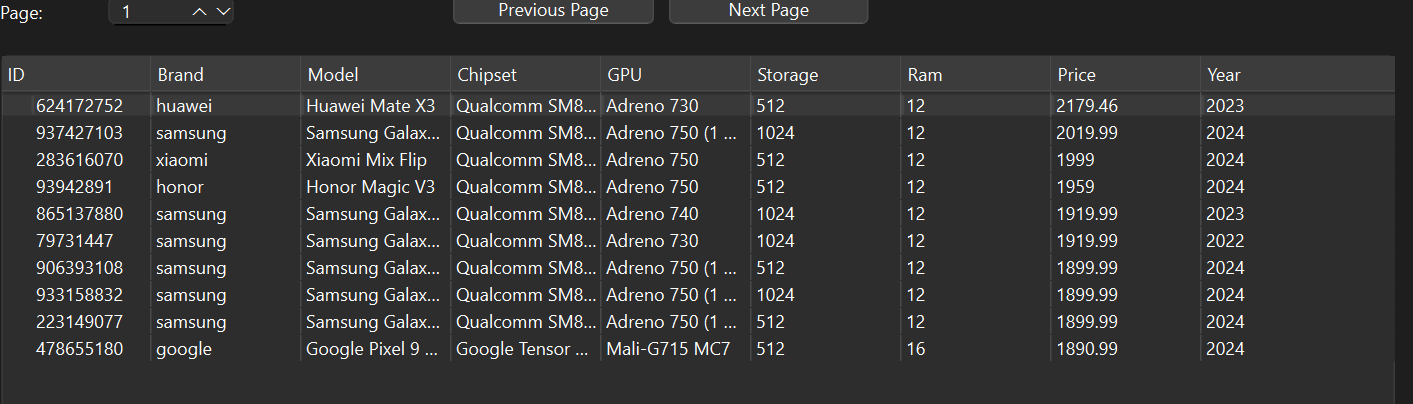
****

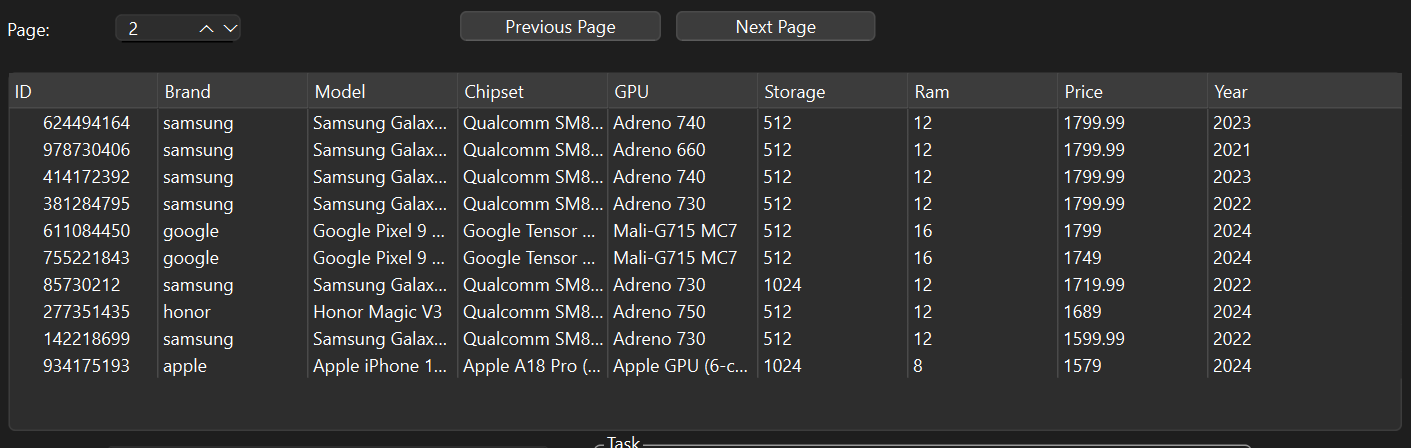
****

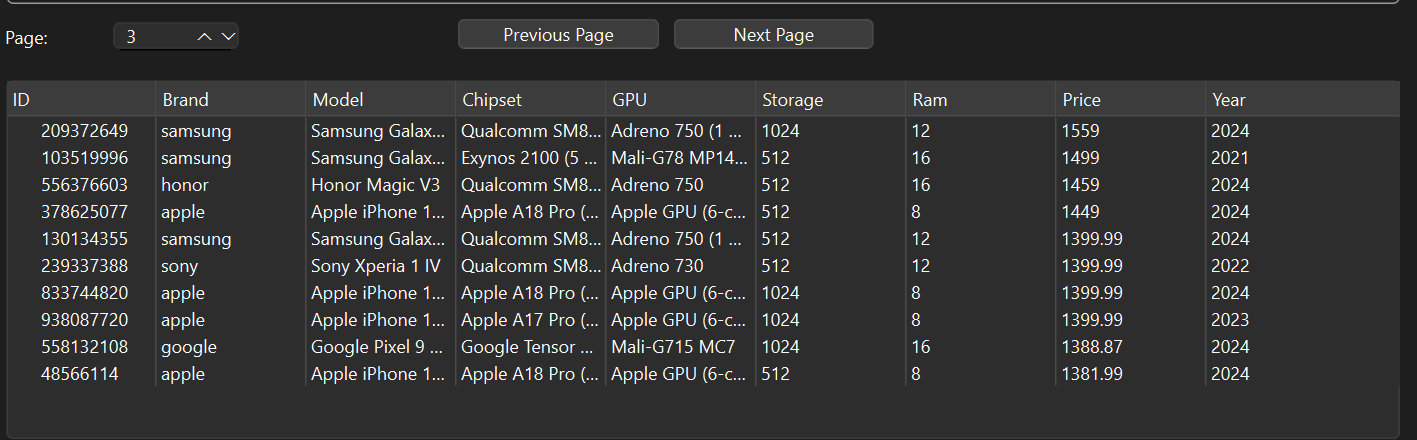
****

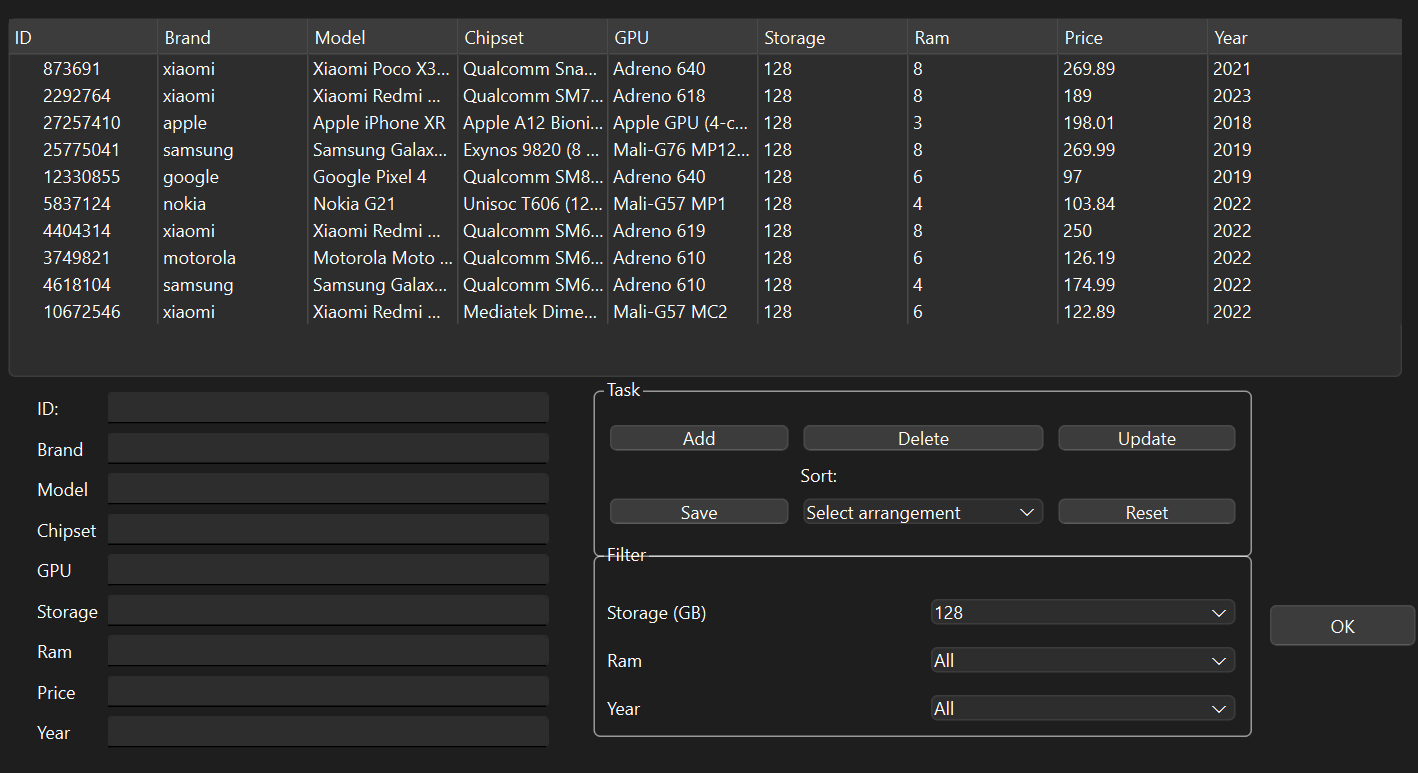
****

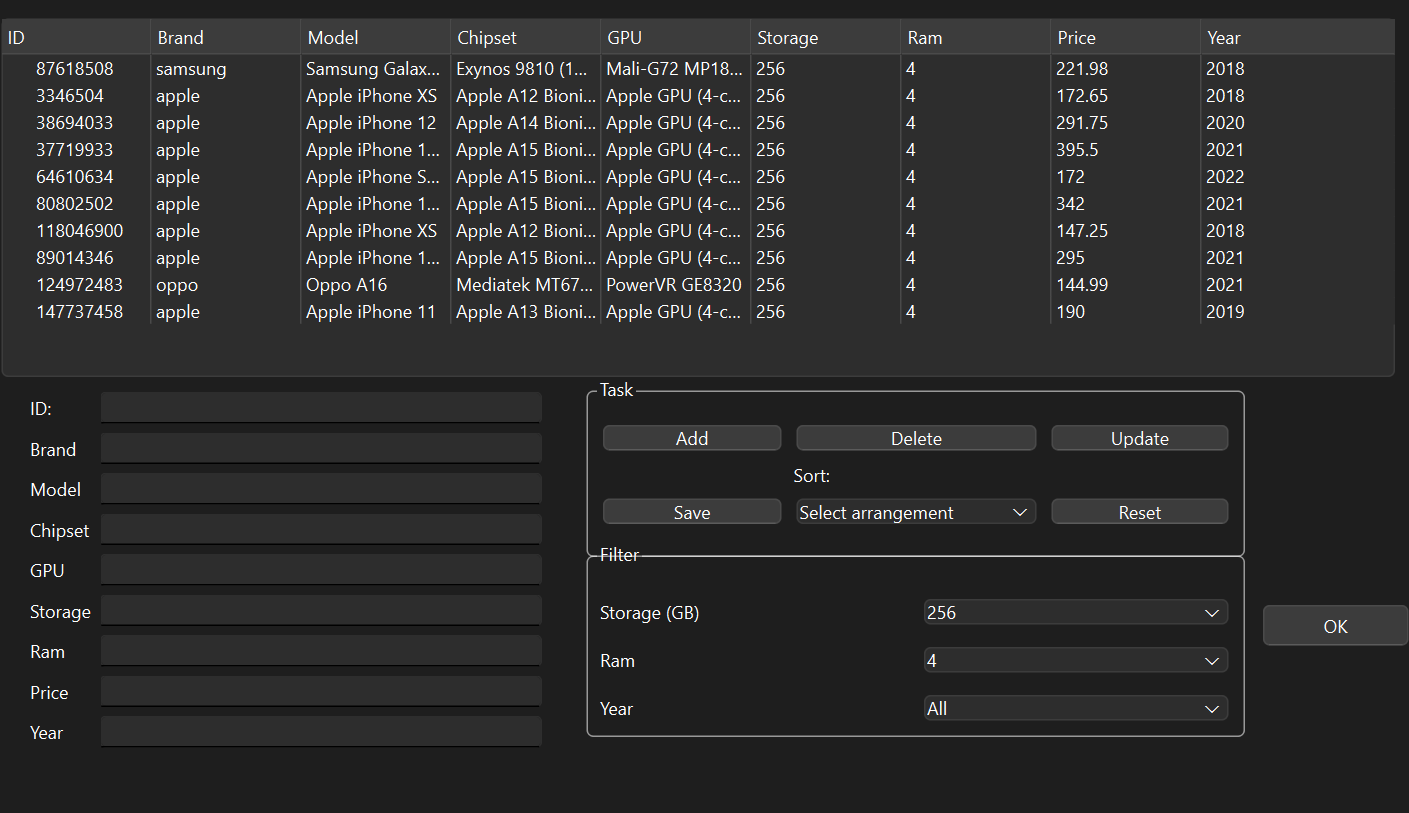
****

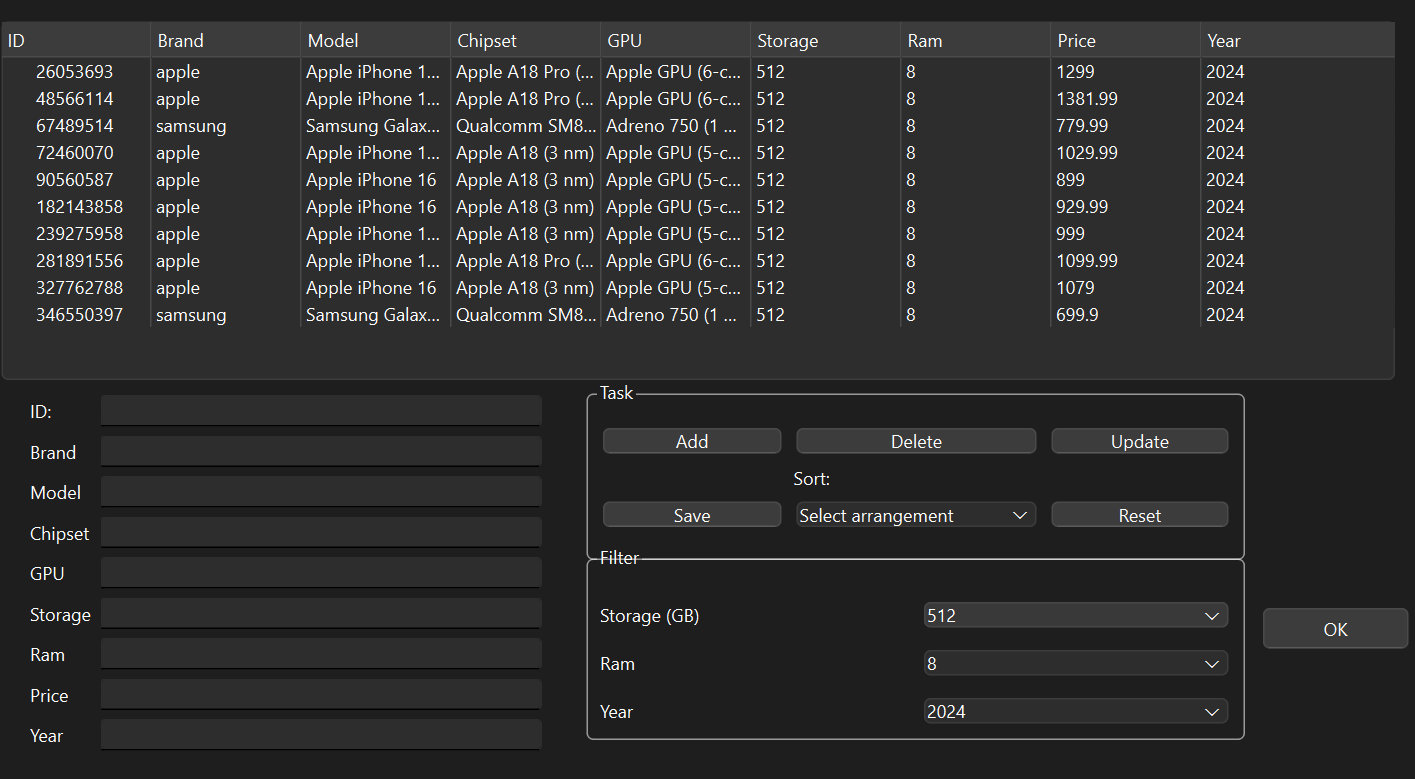
****

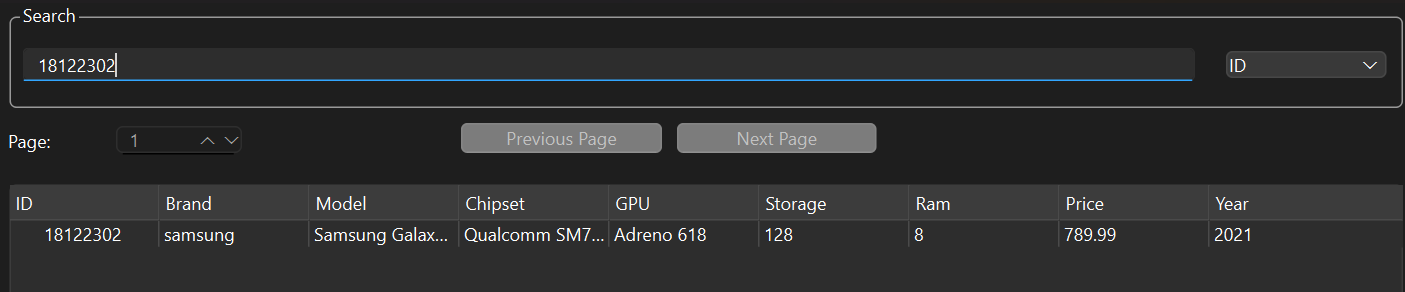
****

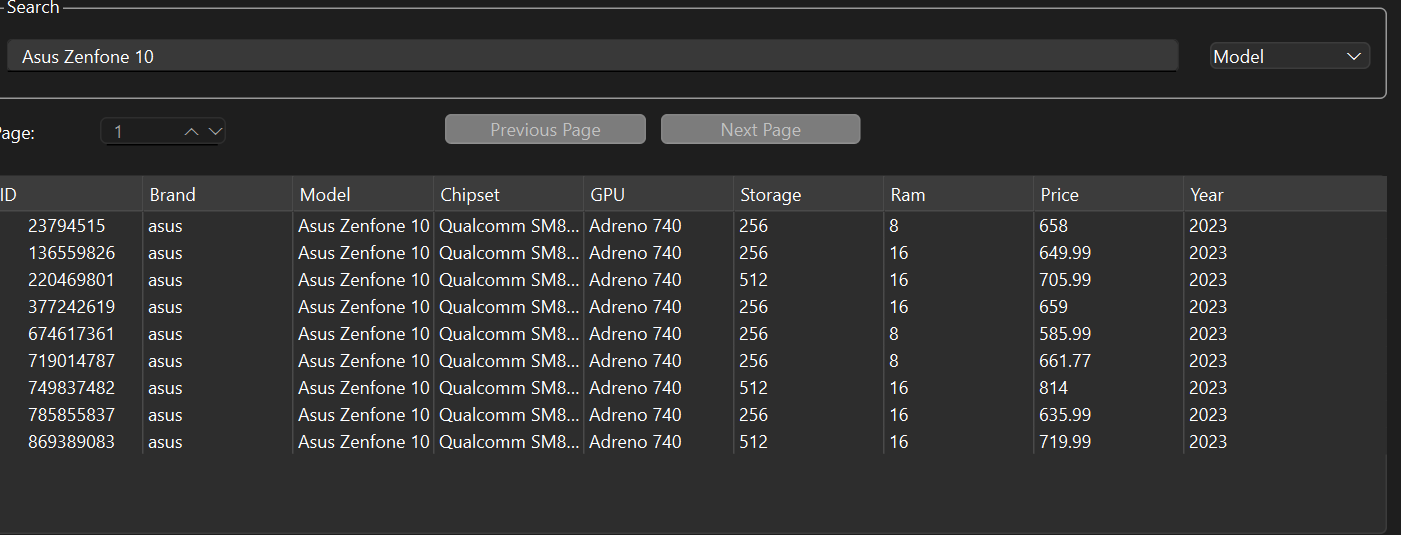
****

****

****

****

****

****