Phone Directory Application

Abstract

We present a phone directory application which is implemented using data structures and file handling. This application helps in creating and managing a contact book. Set of functions are used for inserting, searching, sorting, and deleting operations. Implementation of file handling helps us to save our contacts for future. Our application provides security to the user by assigning passwords.

Introduction

This project demonstrates the working of contact book application using efficient data structure for implementing this in a quick fashion. The data structure used is mainly linked list. We are having one main csv file which will be storing the name and password of those whose contact list is to be created. Each user present in the main csv file will be having their own contact list which can only be accessed by them. We can create a new contact, search ,log into our contact list using our password and delete the contacts. Bubble sort is used for sorting the list in ascending order .
If the user forgets his/her password he/she can log in using the admin password and change their password.

Use of Data Structure

The use the data structures in a phone directory application offers several benefits that not only improves its performance but also enhances the user experience and makes the codebase more maintainable. The choice of data structures aligns with the specific requirements and use cases of the application to provide the best overall benefits. Data structures have allowed us to efficiently store contact information, minimizing memory usage. They provide quick data retrieval and help organize contacts logically.

We have used linked list to insert ,delete and perform other functions . By using linked list we could dynamically allocate our memory which is an important part of our application. Linked lists excel at insertions and deletions

in the middle of the list. We only need to update pointers, the time complexity for these operations is O(1) on average, as opposed to O(n) for arrays where elements may need to be shifted. Linked lists are more memory-efficient than arrays in some cases. They don't require a fixed, contiguous block of memory.

Use of CSV file

CSV file serves an important role in our phonebook application, as it allows us to save, load, and manage contact information, user accounts, and other data.CSV file stores the user account information, usernames and passwords. This data is typically stored in a secure manner and loaded during the login process to authenticate users. When a new contact is added or an existing one is edited, the changes are saved to the corresponding file

Time Complexity

The time complexity of your code depends on the specific operations we perform and the size of the data we are working with. In most cases, the dominant factors are the file I/O operations, sorting algorithms and the number of elements in the linked lists .

In our application loadAccountsCSV function which reads account data from a file has time complexity of O(n), where 'n' is the number of accounts in the file. The loadContactCSV function which reads contact data from a file has time complexity of O(m), where 'm' is the number of contacts in the file. The appendContactNode function appends a contact node to the linked list. Its time complexity is O(1) on average.

The unloadAccountCSVfunction writes account data to a file after sorting it. The sorting part has a time complexity of O(n^2) because it uses a simple bubble sort. The login function involves a loop that allows users to manage their contacts. The loop's complexity depends on the number of operations performed by the user.

The searchContact functionhas a time complexity is O(m), where 'm' is the number of contacts.Functions like newAccount, loggingIn, deleteAccount, newcontact, deletecontact, editName, editNumber, and forgetPassword all have time complexities that depend on the specific operations within them.

These functions involve interactions with linked lists, file operations, and conditional checks.

Data Storage and Security

Our phone directory application provides us with the feature of data storage and security. By the use of CSV file we can store our data for future use. Passwords are assigned so that one user cannot view or edit the contacts of another.