NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY
Faculty of Computing and Informatics
School of Computing
Department of Software Engineering

13 Jackson Kaujeua Street
Private Bag 13388
Windhoek
NAMIBIA

T: +264 61 207 2052
F: +264 61 207 9052
E: dse@nust.na
W: www.nust.na

Data Structure and Algorithms, SEMESTER 2, 2024

MODE OF STUDY: FULL-TIME

SN	Name	Student Number	Specializations	Role played in the project
1	**Elise** Mwandingi	218110510	Informatics	Worked on the project and document
2	Fiina Kornelius	223138436	Informatics	Worked on the project functionalities
3	Haukena Ndapewoshali	224066846	Computer science	Worked on the project
4	Mathew Endjambi	224061720	Computer science	Worked on the project and flowchart
5	Alcanio Tshilumba	224009427	Computer science	Worked on the project And flowchart

Name of Team Leader (TL): Elise Mwandingi

Introduction

This paper serves as a detailed design for a phonebook application specifically made for a Namibian telecommunication company. The program will use simple linear data structures like arrays and linked lists to maintain contacts effectively. Inserting, searching, displaying, deleting, updating, and optionally sorting contacts are among the fundamental features. The algorithms are presented in pseudocode in this document, which also includes a detailed description of every module and function required for implementation.

Pseudocode Representation

```
BEGIN Phonebook Application
    Initialize phonebook with CONTACTS
    WHILE true DO
        Display Menu ()
        choice = GetUserChoice ()

        SWITCH choice
            CASE 1:
                name = GetInput("Enter name:")
                phone = GetInput("Enter phone:")
                InsertContact(phonebook, name, phone)

            CASE 2:
                name = GetInput("Enter name to be searched:")
                phone = SearchContact(phonebook, name)
                IF phone is NOT NULL THEN
```

```
        Print("Phone: " + phone)
    ELSE
        Print("Contact not found.")

CASE 3:
    DisplayAllContacts(phonebook)

CASE 4:
    name = GetInput("Enter name to delete:")
    DeleteContact(phonebook, name)

CASE 5:
    name = GetInput("Enter name to update:")
    newPhone = GetInput("Enter new phone:")
    UpdateContact(phonebook, name, newPhone)

CASE 6:
    SortContacts(phonebook)

CASE 7:
    Print("Exiting...")
    RETURN

DEFAULT:
```

```
            Print("Invalid option. Please try again.")
        END SWITCH
    END WHILE
END PhonebookApplication
```

2. Insert contact

```
FUNCTION InsertContact(phonebook, name, phone)
    IF phonebook.size < MAX_CONTACTS THEN
        FOR EACH contact IN phonebook.contacts DO
            IF contact.name == name THEN
                Print("Contact already exists.")
                RETURN
            END IF
        END FOR
        Append new Contact(name, phone) to phonebook.contacts
        Print("Contact added.")
    ELSE
        Print("Phonebook is full.")
END FUNCTION
```

3.Search contact

```
FUNCTION SearchContact(phonebook, name) RETURNS String
    FOR EACH contact IN phonebook.contacts DO
        IF contact.name == name THEN
```

```
            RETURN contact.phone
        END IF
    END FOR
    RETURN NULL
END FUNCTION
```

4. Display all contacts

```
FUNCTION DisplayAllContacts(phonebook)
    IF phonebook.contacts IS EMPTY THEN
        Print("No contacts available.")
    ELSE
        FOR EACH contact IN phonebook.contacts DO
            Print("Name: " + contact.name + ", Phone: " + contact.phone)
        END FOR
    END IF
END FUNCTION
```

5.Delete Conatct

```
FUNCTION DeleteContact(phonebook, name)
    FOR i FROM 0 TO phonebook.contacts.size - 1 DO
        IF phonebook.contacts[i].name == name THEN
            Remove phonebook.contacts[i]
            Print("Contact deleted.")
            RETURN
        END IF
    END FOR
```

```
    Print("Contact not found.")
END FUNCTION
```

6. update contact

```
FUNCTION UpdateContact(phonebook, name, newPhone)
    FOR EACH contact IN phonebook.contacts DO
        IF contact.name == name THEN
            contact.phone = newPhone
            Print("Contact updated.")
            RETURN
        END IF
    END FOR
    Print("Contact not found.")
END FUNCTION
```

7. sort conatcs (optional)

```
FUNCTION SortContacts(phonebook)
    Sort phonebook.contacts by contact.name
    Print("Contacts sorted.")
END FUNCTION
```

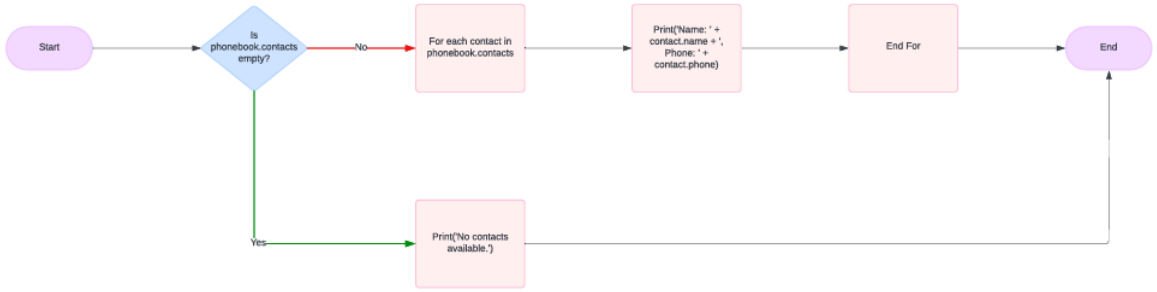
8. efficiently analysis of search algorithm

```
FUNCTION AnalyzeSearchEfficiency()
    Print("Search Algorithm Efficiency: O(n) for linear search.")
```

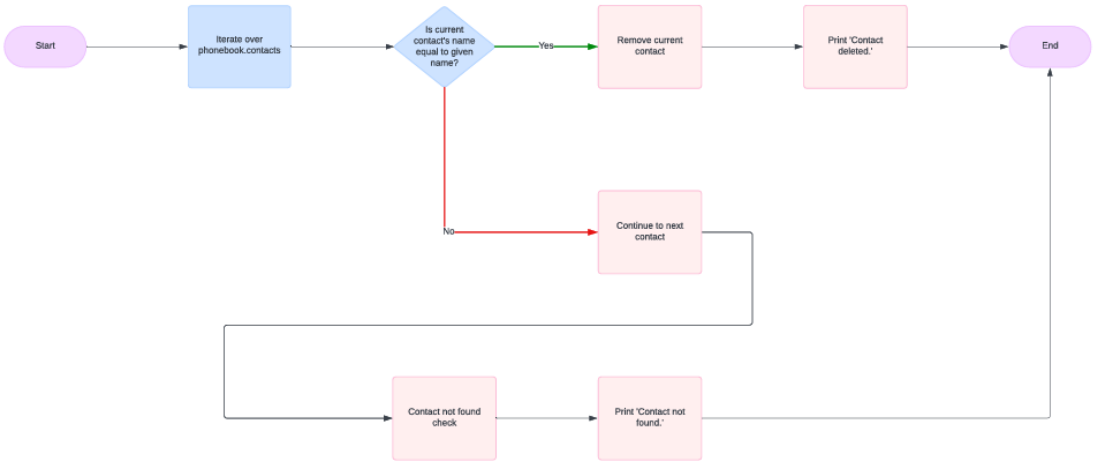
END FUNCTION

FLOW CHART REPRESENTATION

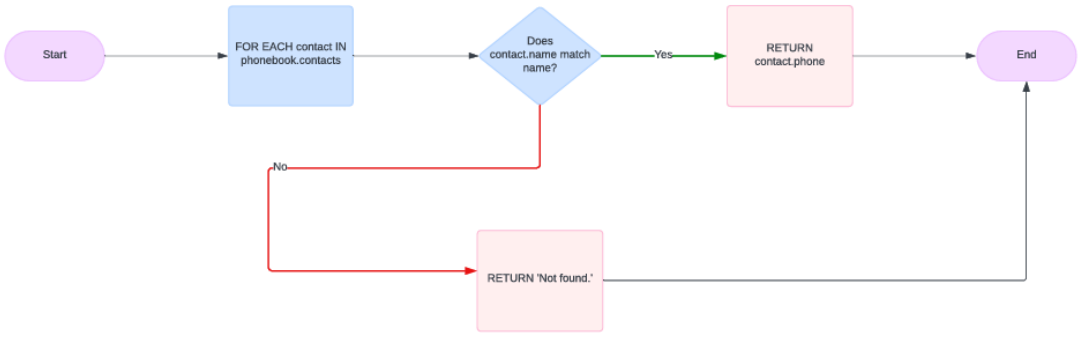
1.Display



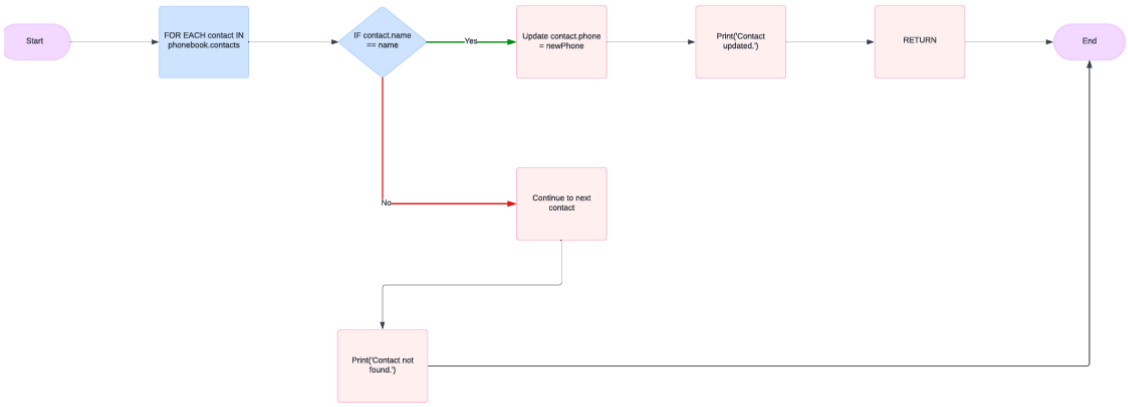
2. Delete



3.Search

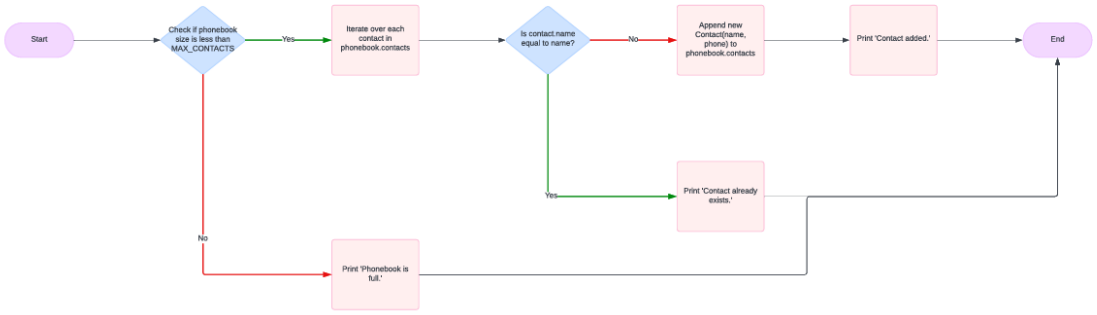


4. Update



5. Sort

6. Insert



7. Analyse

Conclusion

This document outlines the design of a phonebook application using pseudocode representation of its various modules and functions. The application can effectively handle contact management tasks by utilizing simple linear data structures, which qualifies it for use by a telecom provider. This document's well-defined format will help the team collaborate as each member focuses on their own portions of the project, ensuring seamless implementation. For greater efficiency, future improvements can think about incorporating more sophisticated data structures.