**README file**

**PhoneBook Application**

**Overview**

This is a simple Java-based phonebook application where users (specifically an admin) can manage a list of contacts. The admin can add, update, search, display, and delete contacts. Each contact contains several details, including the person's name, mobile number, telephone number, email address, company, department, and position. The data is managed using a doubly linked list for efficient contact insertion and deletion.

**Features**

1. **Admin Login**:
   * Admin must log in with a pre-set username and password to access the phonebook.
   * Username: admin
   * Password: ilovedsa
2. **Add Contact**:
   * Add a new contact with the following details: name, mobile number, telephone number, email address, company, department, and position.
   * Contacts are inserted in alphabetical order based on the name.
3. **Display All Contacts**:
   * View all contacts stored in the phonebook.
4. **Search Contact by Name**:
   * Search for a contact by their name.
5. **Update Contact**:
   * Update a contact’s information (mobile number, telephone number, email address, company, department, and position) based on the contact name.
6. **Delete Contact**:
   * Remove a contact from the phonebook by name.
7. **Exit**:
   * Exit the application safely.

**Classes**

1. **Contact**
   * Represents a single contact in the phonebook.
   * Contains fields such as name, mobileNumber, telephoneNumber, emailAddress, company, department, position.
   * Methods:
     + displayContact(): Displays the details of a contact.
2. **PhoneBook**
   * Manages the collection of contacts using a doubly linked list.
   * Methods:
     + addContact(): Adds a contact in alphabetical order.
     + displayAllContacts(): Displays all contacts.
     + searchContactByName(): Searches for a contact by name.
     + updateContact(): Updates an existing contact.
     + deleteContact(): Deletes a contact by name.
3. **PhoneBookApplication**
   * Main class that provides a user interface (console-based).
   * Implements the admin sign-in and a menu-driven interface to perform various operations.

**How to Use**

1. **Run the Program**:
   * Compile and run the PhoneBookApplication class.
2. **Admin Sign-In**:
   * The program will prompt for admin credentials. Use the following credentials to sign in:
     + Username: admin
     + Password: ilovedsa
   * If the credentials are incorrect, the program will exit.
3. **Phonebook Operations**:
   * After successful login, the menu with options will be displayed.
   * You can perform the following operations:
     + Add a new contact.
     + Display all contacts.
     + Search for a contact by name.
     + Update an existing contact.
     + Delete a contact.
     + Exit the program.
4. **Exit**:
   * Choose the Exit option (6) from the menu to close the program.

**Sample Interaction**

Admin Sign In

Enter username: admin

Enter password: ilovedsa

Sign in successfully.

--- Welcome to Phone Book Menu ---

1. Add Contact

2. Display All Contacts

3. Search Contact by Name

4. Update Contact

5. Delete Contact

6. Exit

Choose an option: 1

Enter name: Hamutenya Abraham

Enter mobile number: 1234567890

Enter telephone number: 061344567

Enter email address: hamutenyaabraham@example.com

Enter company: Telecom Namibia

Enter department: IT

Enter position: Manager

Contact saved successfully.

--- Welcome to Phone Book Menu ---

Choose an option: 2

Name: Hamutenya Abraham

Mobile Number: 1234567890

Telephone Number: 061344567

Email Address: 061344567

Company: Telecom Namibia

Department: IT

Position: Manager

---------------------------------

--- Welcome to Phone Book Menu ---

Choose an option: 6

Exiting...

**Requirements**

* Java Development Kit (JDK) 8 or later.

**How to Compile and Run**

1. **Compile**:

javac -d . PhoneBookApplication.java

1. **Run**:

java phonebookapplication.PhoneBookApplication

**Future Enhancements**

* Implement persistence by saving contacts to a file.
* Add user roles with different access levels.
* Implement a graphical user interface (GUI).

**Author**

* Developed by [Hamutenya Abraham]