```
import json

class Contact:
    def __init__(self, name, phone, email, address):
        self.name = name
        self.phone = phone
        self.email = email
        self.address = address

    def __str__(self):
        return f"Name: {self.name}, Phone: {self.phone}, Email: {self.email}, Address: {self.address}"

class ContactManager:
    def __init__(self):
        self.contacts = self.load_contacts()

    def load_contacts(self):
        try:
            with open('contacts.json', 'r') as file:
                contacts_data = json.load(file)
                return [Contact(**contact) for contact in contacts_data.get('contacts', [])]
        except FileNotFoundError:
            return []

    def save_contacts(self):
        contacts_data = {'contacts': [vars(contact) for contact in self.contacts]}
        with open('contacts.json', 'w') as file:
            json.dump(contacts_data, file, indent=2)

    def display_contacts(self):
        if not self.contacts:
            print("No contacts found.")
        else:
            print("Contact List:")
            for contact in self.contacts:
                print(contact)

    def add_contact(self):
        name = input("Enter contact name: ")
        phone = input("Enter contact phone number: ")
        email = input("Enter contact email: ")
        address = input("Enter contact address: ")

        new_contact = Contact(name, phone, email, address)
        self.contacts.append(new_contact)
        self.save_contacts()
        print(f"Contact '{name}' added successfully!")

    def search_contact(self, search_term):
        results = [contact for contact in self.contacts if
                   search_term.lower() in contact.name.lower() or search_term in contact.phone]
        return results

    def update_contact(self, contact_to_update):
        print(f"Updating contact: {contact_to_update}")
        contact_to_update.phone = input("Enter new phone number: ")
        contact_to_update.email = input("Enter new email: ")
        contact_to_update.address = input("Enter new address: ")
        self.save_contacts()
        print("Contact updated successfully!")

    def delete_contact(self, contact_to_delete):
        self.contacts.remove(contact_to_delete)
        self.save_contacts()
        print(f"Contact '{contact_to_delete.name}' deleted successfully!")

def main():
    contact_manager = ContactManager()

    while True:
        print("\nContact Manager")
        print("1. View Contact List")
        print("2. Add Contact")
        print("3. Search Contact")
        print("4. Update Contact")
        print("5. Delete Contact")
        print("6. Exit")

        choice = input("Enter your choice (1-6): ")

        if choice == '1':
            contact_manager.display_contacts()
        elif choice == '2':
            contact_manager.add_contact()
        elif choice == '3':
            search_term = input("Enter name or phone number to search: ")
            results = contact_manager.search_contact(search_term)
            if results:
                print("Search Results:")
                for result in results:
                    print(result)
            else:
                print("No matching contacts found.")
        elif choice == '4':
            search_term = input("Enter name of the contact to update: ")
            results = contact_manager.search_contact(search_term)
            if results:
                contact_manager.update_contact(results[0])
            else:
                print(f"Contact with name '{search_term}' not found.")
        elif choice == '5':
            search_term = input("Enter name of the contact to delete: ")
            results = contact_manager.search_contact(search_term)
            if results:
                contact_manager.delete_contact(results[0])
            else:
                print(f"Contact with name '{search_term}' not found.")
        elif choice == '6':
            print("Exiting the Contact Manager. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 6.")

if __name__ == "__main__":
    main()
```