```python
from os.path import exists

#   Initialize CONTACTS as an empty array
CONTACTS = []


# A function to reduce redundant code
def invalid_error():
    print("Invalid Input")


#   A function to take confirmation from user
def is_confirmed(text):
    decision = input(f"{text} (y,n) : ")
    if decision.lower() == "y":
        return True
    return False


#   Initialize contacts from a text file to the program
def init():
    if exists("PhoneBook/data.txt"):
        data_file = open("PhoneBook/data.txt", "r")
        data = data_file.readlines()
        for line in data:
            if len(line.split("|")) >= 3:
                name, address, email, *phones = line.strip().split("|")
                CONTACTS.append(
                    {
                        "name": name,
                        "address": address,
                        "email": email,
                        "phones": phones,
                    }
                )
        data_file.close()


#   Function to save all contacts in a text file
def save_file():
    data_file = open("PhoneBook/data.txt", "w")
    data_file.writelines(
        f'{("|".join((str(item[key]) for key in item if key !=
"phones")))}|{"|".join(phone for phone in item["phones"])}\n'
        for item in CONTACTS
    )
    data_file.close()


#   Take input for a contacts on adding and updating
def take_contact_inputs():
    name = input("Name : ")
    address = input("Address : ")
    email = input("Email : ")
    phones = input("Phone (Comma Separated) : ").split(",")
    return {"name": name, "address": address, "email": email, "phones": phones}


#   Function to add new contact
def add_contact():
    print("New Contact :")
    CONTACTS.append(take_contact_inputs())
```

```
    print_all_contacts()
    if is_confirmed("Add again ?"):
        add_contact()


#   Function to update an existing contact
def update_contact():
    print_all_contacts()
    option = int(input("Enter a number to update : "))
    if option <= len(CONTACTS):
        CONTACTS[option - 1] = take_contact_inputs()
        print_all_contacts()
        if is_confirmed("Update again?"):
            remove_contact()
    else:
        print("Invalid input")


#   Print contacts from arguments to the console
def print_contacts(contacts):
    if len(contacts) == 0:
        print("\n\nNo contact found\n\n")
        return
    index = 0
    for contact in contacts:
        index += 1
        print("\n", index, " .")
        print("Name : ", contact["name"])
        print("Address : ", contact["address"])
        print("Email : ", contact["email"])
        print("Phones : ", ", ".join(contact["phones"]), "\n")


#   Print all contacts to the console
def print_all_contacts():
    print_contacts(CONTACTS)


#   Search for a contact
def search_contact(query):
    results = []
    for item in CONTACTS:
        if (
            item["name"].lower().__contains__(query.lower())
            or item["address"].lower().__contains__(query.lower())
            or item["email"].lower().__contains__(query.lower())
        ):
            results.append(item)
        else:
            for phone in item["phones"]:
                if phone.lower().__contains__(query.lower()):
                    results.append(item)
    return results


#   Function to print search result to the console
def show_search_result():
    query = input("Search for contact : ")
    print_contacts(search_contact(query))


#   Function to remove exisintg contact
```

```python
def remove_contact():
    print_all_contacts()
    option = int(input("Enter a number to remove : "))
    if option <= len(CONTACTS):
        del CONTACTS[option - 1]
        print_all_contacts()
        if is_confirmed("Delete again?"):
            remove_contact()
    else:
        print("Invalid input")


#   Main function containing the menu of the application
def main():
    init()
    while True:
        print(
            """
        Welcome to Contact List

        1. View Contacts
        2. Add New Contact
        3. Update Contact
        4. Remove Contact
        5. Search Contact
        0. Exit
        """
        )
        option = int(input("Enter an option : "))
        if option == 0:
            break
        menus = {
            1: print_all_contacts,
            2: add_contact,
            3: update_contact,
            4: remove_contact,
            5: show_search_result,
        }
        if menus.keys().__contains__(option):
            menus[option]()
        else:
            print("Invalid Option")
    save_file()


main()
```