**Exercise 8**

**Aim**

To create a Contact Management System that allows users to add, search, and display contact information, using exception handling to manage input errors effectively.

**Algorithm**

|  |  |  |
| --- | --- | --- |
| Step 1 | **:** | Start the Program. |
| Step 2 | **:** | Create the Contact Class for storing name, phone number, and email. |
| Step 3 | **:** | Create the ContactManager Class for managing contacts |
| Step 4 | **:** | Get User Input for menu selection. |
| Step 5 | **:** | Execute user choice for contacts management. |
| Step 6 | **:** | Handle exceptions in user input |
| Step 7 | **:** | Display the Result |
| Step 8 | **:** | Stop the Program. |

**Program:**

class Contact:

def \_\_init\_\_(self, name, phone\_number, email):

self.name = name

self.phone\_number = phone\_number

self.email = email

def \_\_str\_\_(self):

return f"Name: {self.name}, Phone: {self.phone\_number}, Email: {self.email}"

class ContactManager:

def \_\_init\_\_(self):

self.contacts = []

def add\_contact(self, name, phone\_number, email):

contact = Contact(name, phone\_number, email)

self.contacts.append(contact)

print(f"Contact '{name}' added successfully.")

def search\_contact(self, name):

for contact in self.contacts:

if contact.name.lower() == name.lower():

return contact

return None

def display\_contacts(self):

if not self.contacts:

print("No contacts available.")

return

print("\nContact List:")

for contact in self.contacts:

print(contact)

def main():

manager = ContactManager()

while True:

print("\nContact Management Menu:")

print("1. Add Contact")

print("2. Search Contact")

print("3. Display All Contacts")

print("4. Exit")

choice = input("Choose an option (1-4): ")

try:

if choice == '1':

name = input("Enter the contact's name: ")

phone\_number = input("Enter the contact's phone number: ")

email = input("Enter the contact's email: ")

manager.add\_contact(name, phone\_number, email)

elif choice == '2':

name = input("Enter the name of the contact to search: ")

contact = manager.search\_contact(name)

if contact:

print(contact)

else:

print(f"Contact '{name}' not found.")

elif choice == '3':

manager.display\_contacts()

elif choice == '4':

print("Exiting the contact management system.")

break

else:

print("Invalid choice. Please select a valid option.")

except Exception as e:

print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Output**

**Result**

The Contact Management System was successfully developed, enabling easy contact management with an effective error handling mechanism.