

**LAB ASSIGNMENT 2**

|  |  |
| --- | --- |
| Name | Muhammad Eman |
| Course Title | OOP |
| Date | 11/10/2024 |
| Reg no | SP24-BSE-071 |
| Submitted to | Shahid Bhatti |

**Classes**

1. **Main**

In main class I have created and interface to interact with different classes. You can manage contacts chat with someone and in chat you can select whether to send messages, read messages, or delete messages. It have different loops to give an experience of a fullfilled chatting application.

1. **Sms**

SMS class stores different attributes of an SMS for example it handles centre number receiver number content status of the message and whether the message is read or unread. It also handles the time at which the messages sent and stores messages along with date and time.

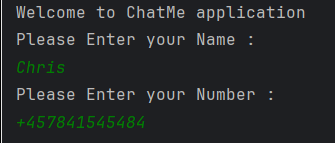
1. **Contacts**

The contact class deals with the attributes of a contact. It handles contact name number and contact ID.

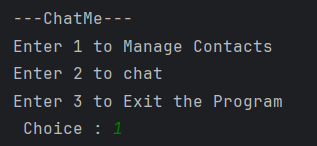
1. **ChatMe**

This class store the back end of a chatting system it has methods for sending messages, reading messages and deleting messages. It also has methods for managing contacts. It has a functionality to save contact delete contact and search contact

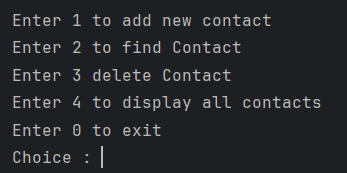
screenshots of output screen



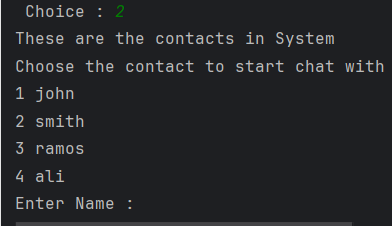
At first it asks for user Name and number

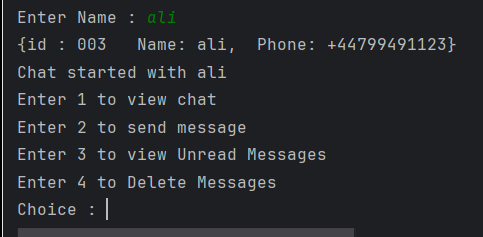


Then this menu appear which has listed options , user can select any option they want.



Contact managing has these functionalities





Chat has these functionalities

**Methods Used**

**1. setSenderName(String senderName):** Sets the sender's name.

**2. getSenderName():** Returns the sender's name.

**3. setSenderNumber(String senderNumber):** Sets the sender's phone number.

**4. getSenderNumber():** Returns the sender's phone number.

**5. sendMessage(String name, int contactIndex):** Sends a message to a contact by asking the user for the message content. The user types messages until "//" is entered, and the messages are stored in `allSms[]`.

**6. viewUnreadMessages(String receiverName):** Views unread messages from a specific contact, marking them as read. Messages are retrieved and sorted.

**7. deleteMessage():** Deletes a specific message from `allSms[]` by asking for an index.

**8. chatOverview(String receiverName):** Displays all messages with a contact without changing their read/unread status.

**9. manageContacts():** A menu that allows the user to add, find, delete, or display contacts. The user selects the action via input.

**10. addContact():** Adds a new contact by asking for a name and phone number.

**11. findContactWithName(String name):** Finds a contact by name and returns `true` if found, otherwise returns `false`.

**12. getContactIndexWithName(String name):** Returns the index of a contact based on the name.

**13. getContactNumberWithName(String name):**

Retrieves the phone number of a contact based on the name.

**14. findContact():** Asks the user for a contact name and searches for it using `findContactWithName()`.

**15. deleteContact():** Deletes a contact from `contactList[]` by name, adjusting the array to remove the deleted contact.

**16. displayContacts():** Displays all contacts in `contactList[]`.

**17. getContactForChat():** Prompts the user to select a contact to chat with, returning the name of the selected contact.