# CONTACT MANAGER

Group No. 15

## TEAM MEMBERS

22BT04009 - Arnav Das
22BT04011 - Sammit Bahulekar
22BT04021 - Deepak Sharma

## ABSTRACT

This Java program is a simple contact manager that allows users to add contacts, view all contacts, search for a contact by name, and exit the program. It utilizes serialization to save and load contacts to/from a file named "contacts.dat."

Let's go through the main components and the flow of the program:

### Contact Class

Contact is a simple class representing a contact with name and phoneNumber attributes. It implements Serializable to allow instances of this class to be serialized and stored in a file.

### Serialization

Serialization is used to save and load the contacts from a file. The Contact class implements Serializable, allowing its objects to be serialized and stored in the file. The program uses ObjectOutputStream and ObjectInputStream to achieve this.

### ContactManager Class

ContactManager is the main class containing the main method. It has a constant FILE_NAME representing the name of the file where contacts will be stored. It has an ArrayList<Contact> named contacts to store the contact objects. A Scanner object named scanner is used to take user input.

### loadContactsFromFile

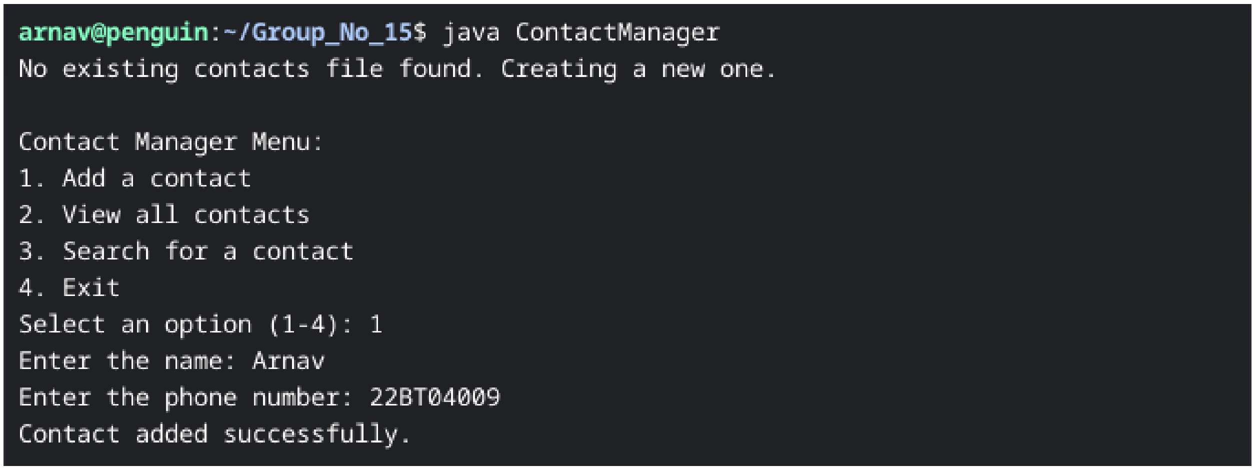
This method attempts to load contacts from the file specified by FILE_NAME. If the file exists, it reads the ArrayList<Contact> from the file and assigns it to the contacts variable. If the file doesn't exist or an error occurs during the process, it prints a message indicating that no existing contacts file was found, and a new one will be created.

### main Method

The main method is the entry point of the program. It starts by calling loadContactsFromFile to load existing contacts. It then enters a loop where it displays a menu to the user and performs actions based on the user's choice until the user chooses to  exit. Depending on the user's choice, it calls the corresponding method (addContact, viewContacts, searchContact, or exits the program).

### addContact

This method prompts the user to enter a name and phone number and creates a new Contact object. The new contact is then added to the contacts ArrayList.

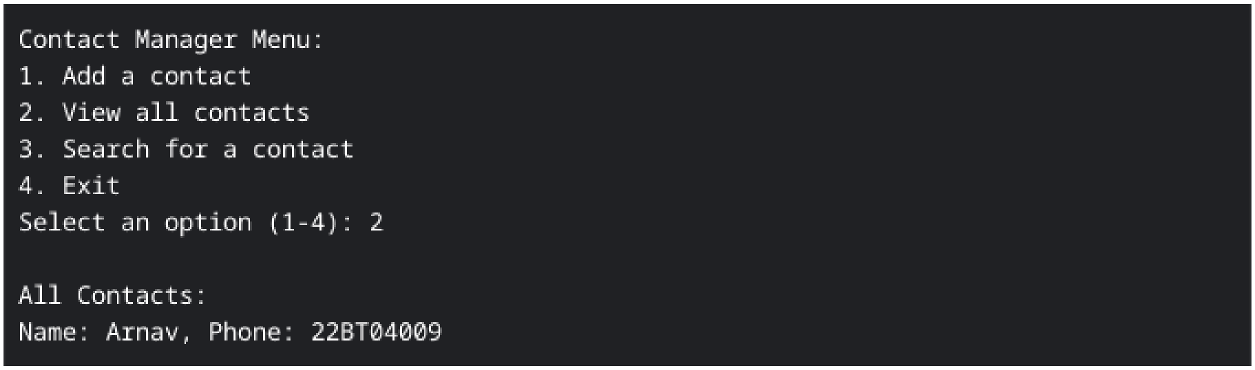


```
arnav@penguin:~/Group_No_15$ java ContactManager
No existing contacts file found. Creating a new one.

Contact Manager Menu:
1. Add a contact
2. View all contacts
3. Search for a contact
4. Exit
Select an option (1-4): 1
Enter the name: Arnav
Enter the phone number: 22BT04009
Contact added successfully.
```

## viewContacts

This method prints all contacts in the contacts ArrayList. If there are no contacts, it prints a message indicating that no contacts were found.

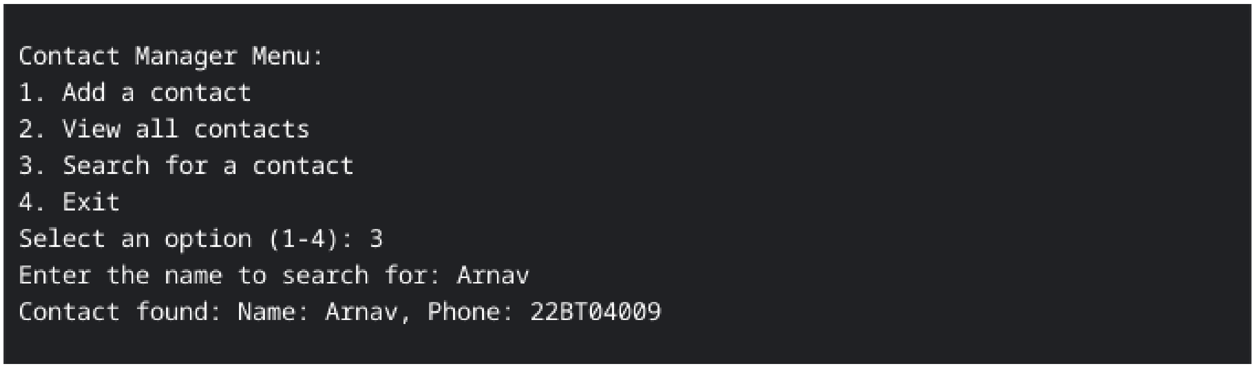


```
Contact Manager Menu:
1. Add a contact
2. View all contacts
3. Search for a contact
4. Exit
Select an option (1-4): 2

All Contacts:
Name: Arnav, Phone: 22BT04009
```

## searchContact

This method prompts the user to enter a name to search for. It then iterates through the contacts and prints the details of the contact if a match is found. If no match is found, it prints a message indicating that the contact was not found.



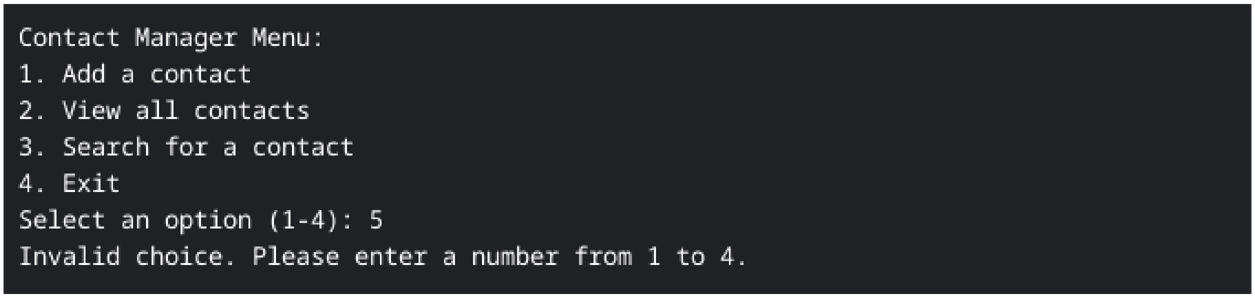
```
Contact Manager Menu:
1. Add a contact
2. View all contacts
3. Search for a contact
4. Exit
Select an option (1-4): 3
Enter the name to search for: Arnav
Contact found: Name: Arnav, Phone: 22BT04009
```

## saveContactsToFile

This method saves the current contacts to the file specified by FILE_NAME.
It uses an ObjectOutputStream to serialize and write the contacts ArrayList to the file.

## Exception Handling

The program uses exception handling to catch InputMismatchException when the user enters something other than a number. It also catches IOException and ClassNotFoundException during file operations.



```
Contact Manager Menu:
1. Add a contact
2. View all contacts
3. Search for a contact
4. Exit
Select an option (1-4): 5
Invalid choice. Please enter a number from 1 to 4.
```