Record Class:

• This class represents a single record in the phone book. Each

record has fields for "firstName", "lastName", "address".

city, and phoneNumber".

The class has getters for each field and overrides the

toString() method to provide a formatted representation of the record.

Phonebook2023 Class:

• This class is responsible for managing the phone book records.

It has a constant FILE NAME that represents the name of the file used to store the records.

The class contains a 'List<Record> named 'records'

to store all the records.

Constructor:

Phonebook2023

• Initializes the records list and attempts to load records from the file specified by 'FILE NAME using the

loadRecordsFromFile()

method.

4. addRecord (Record record) Method: Adds a new record to the phone book.

• It checks if the phone number already exists in the records to

avoid duplicate entries.

If the phone number does not exist,

it adds the new record to the list and saves the records to the file using the'saveRecordsToFile() method.

5. deleteRecord (String phoneNumber) Method:

• Deletes a record based on the given phone number.

It searches for the record with the provided phone number,

and if found, it removes it from the list and saves the updated records to the file.

6 modifyRecord (String phoneNumber) Method:

Modifies a record based on the given phone number.

It searches for the record with the provided phone number, and if found, it prompts the user to enter new information (First Name, Last Name, Address, City) to update the record.

Then it removes the old record and adds the updated record to the list, saving the changes to the file.

7. searchRecords() Method:

Allows the user to search for records based on various criteria

(Last Name, City, Phone Number).

It prompts the user for a search option, performs the search, and displays the matching records.

8 displayAllRecords () Method:

Displays all the records in a grid format with three columns.

• It calculates the number of rows needed to display all the records and then iterates through the records to print them in the grid format.

9 countRecords () Method:

Returns the total number of records in the phone book.

10. Private Helper Methods:

searchByLastName (String lastName), searchByCity (String

city), and searchByPhoneNumber(String phoneNumber) are private methods used by "searchRecords () to perform the actual

search operations based on user input.

findRecordByPhoneNumber(String phoneNumber) is a private

method that helps to find a record by phone number.

1. File Handling Methods:

'loadRecordsFromFile() reads records from the file specified by

FILE NAME and populates the records list.

"saveRecordsToFile() writes the records in the "records list to the file specified by "FILE NAME

main(String[] args) Method:

This is the entry point of the phone book system.

It creates an instance of "Phonebook2023 and starts an interactive

loop, presenting a menu to the user for various actions (Add, Delete,

Modify, Search, Display, Count, Quit). Based on the user's choice, the corresponding methods of Phonebook2023 are called to perform the desired operation.

Overall, this Java code represents a basic phone book system that allows users to manage records efficiently.

The records are stored in a file, and any changes made to the phone book are persisted between different program executions\.