[**Phone Book System Documentation**](#_fgpthda1cu7f) **1**

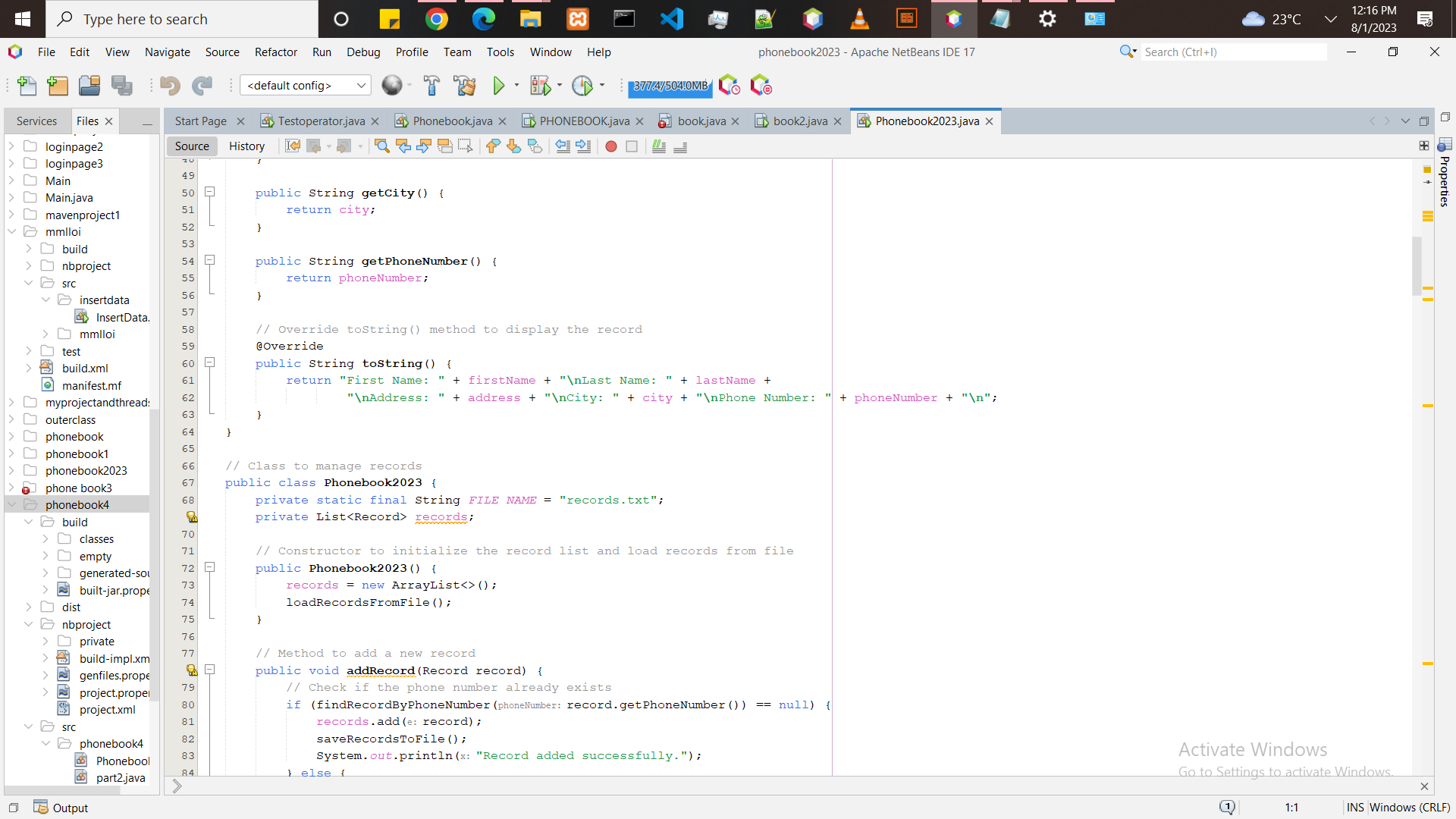
[**Methods:**](#_inn8t0r0d4hl) **2**

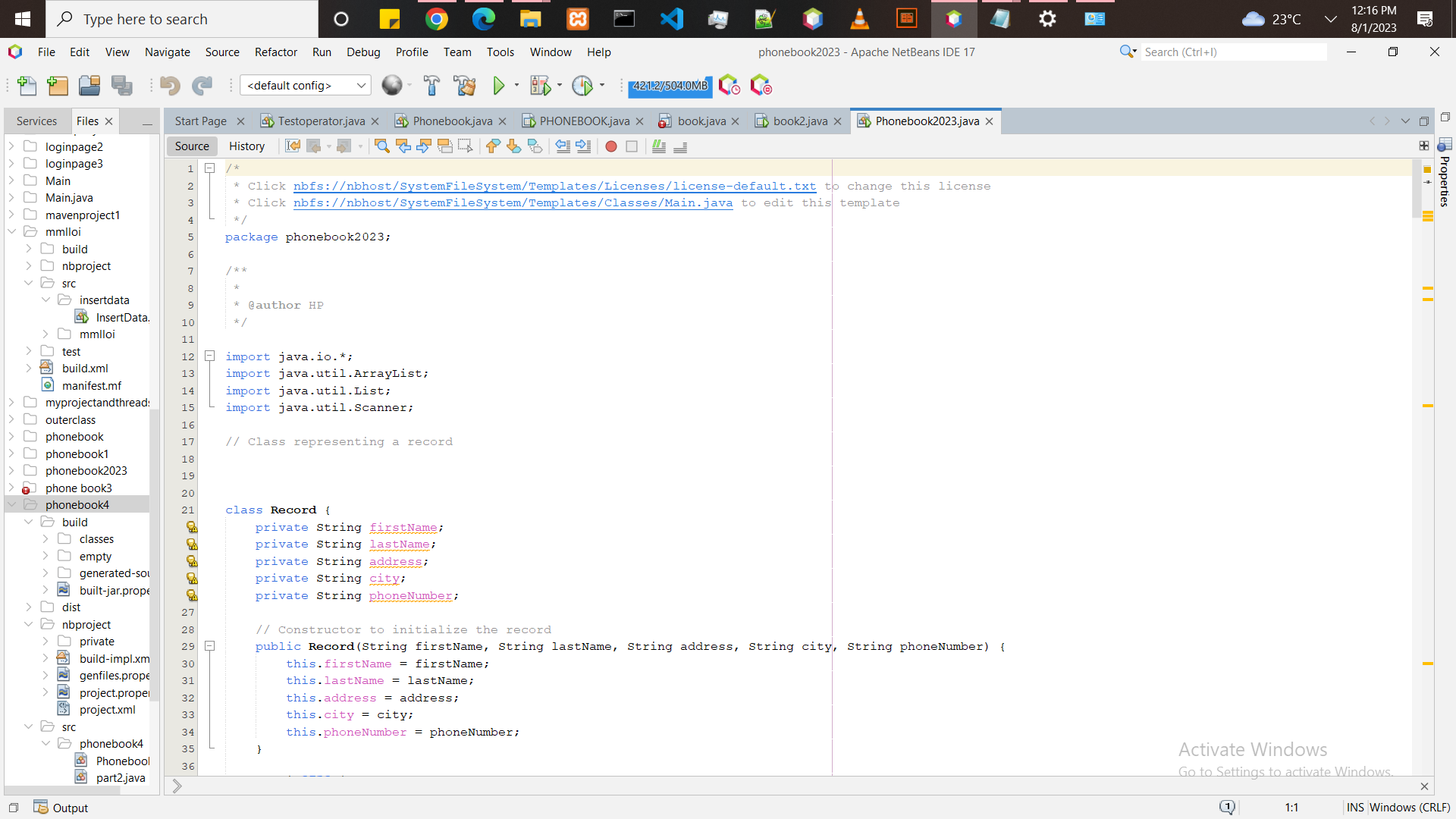
[**METHODS:**](#_sfmy5s2iemz1) **3**

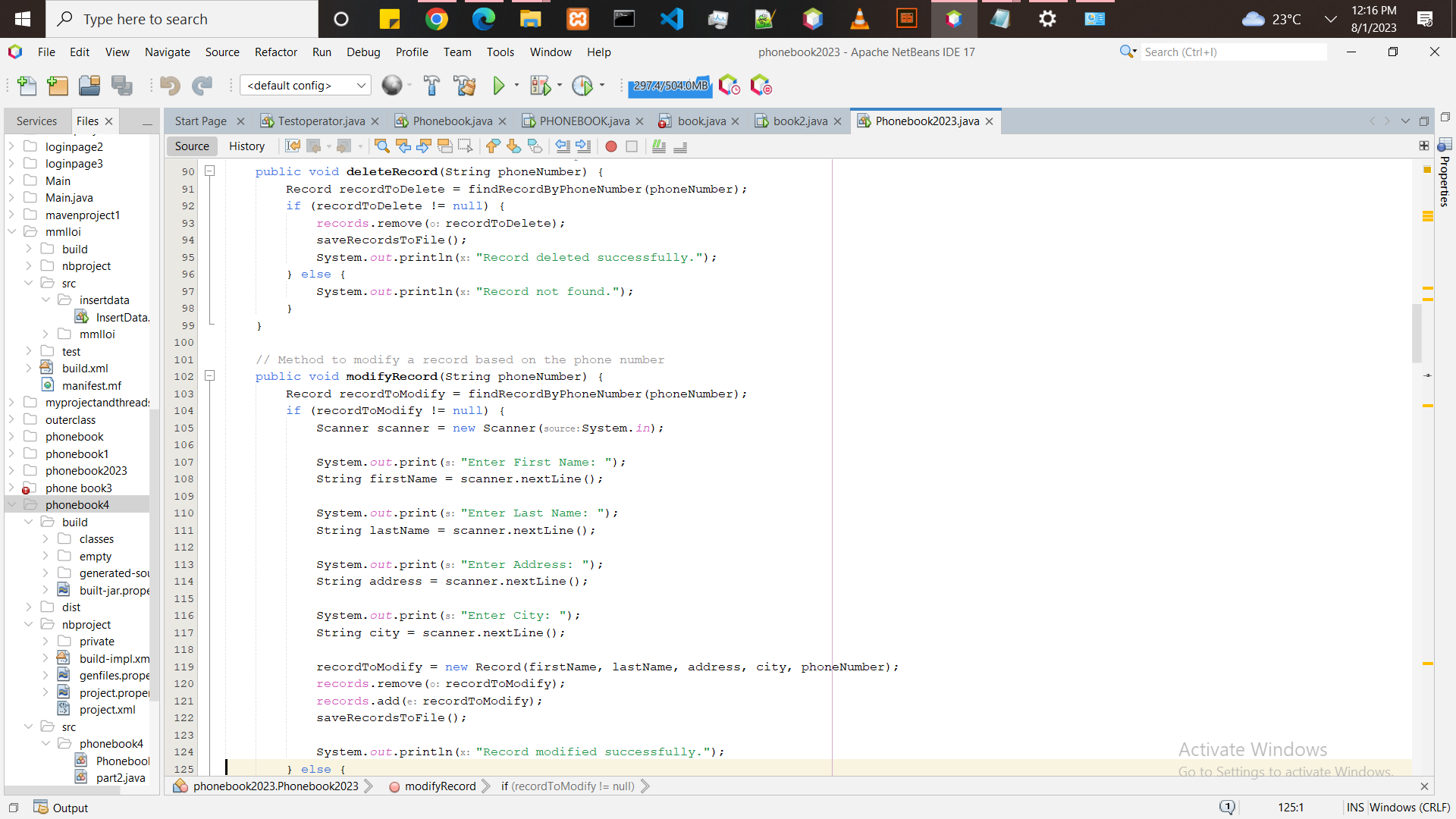
[**3. main() Method**](#_maqc8dsk4k8r) **4**

[**Usage**](#_g22aepn189i5) **5**

[**Conclusion**](#_ut6vysalumaf) **6**







# Phone Book System Documentation

The Phone Book System is a Java program that provides basic functionality to manage a phone book.

Users can perform various operations such as adding, deleting, modifying, searching, displaying. and counting records.

The records are stored in a file, and the program loads the

records from the file during initialization and saves the updated record back to the file after any modification.

Classes

1. Record Class

This class represents a single record in the phone book. Each record contains the following fields:

•"first Name": The first name of the person associated with the record.

•last Name: The last name of the person associated with the record.

• "address": The address of the person associated with the record.

• "city": The city of the person associated with the record.

•"phone Number: The phone number of the person associated with the record.

# Methods:

•"Record(String first Name, String last Name, String address, String city, String phone Number): Constructor to initialize a new record object with the given information.

• “get First Name(): Getter method to retrieve the first name of the record.

• “get Last Name(): Getter method to retrieve the last name of the record.

record.

• “get Address(): Getter method to retrieve the address of the record.

• “get City(): Getter method to retrieve the city of the record.

• “get Phone Number(): Getter method to retrieve the phone number of the record.

• "to String(): Override method to display the record's information in a user-friendly format.

2. Phonebook2023 Class

This class manages the phone book and contains methods to perform various operations on records.

Fields: private static final String FILE\_NAME: The name of the file used to store records.

# METHODS:

• “Phonebook2023(): Constructor to initialize the phone book system. It creates an empty list of records and loads records from the file.

• “add Record (Record record): Method to add a new record to the phone book. Checks for duplicate phone numbers before adding the record.

• “delete Record (String phone Number): Method to delete a record based on the provided phone number.

• “modify Record(String phone Number): Method to modify a record based on the provided phone number.

• “search Records(): Method to search for records based on various criteria (last name city, phone number).

• “display All Records (): Method to display all records in a grid format

• “count Records(): Method to count the total number of records in the phone book.

•”search By Last Name (String last Name): Private method to search

records by last name.

•”search by city(String city): Private method to search records by city.

•”search By Phone Number(String phone Number): Private method to search records by phone number.

• "display Search Results (List<Record> search Results): Private method to display search results.

• "find Record By Phone Number(String phone number): Private

method to find a record by phone number.

•"load Records From File(): Private method to load records from the

file during program initialization.

•”save Records To File(): Private method to save records to the file after any modification.

# 3. Main() Method

The main() method is the entry point of the program. It creates an instance of "Phonebook2023 and provides a menu-driven interface for users to interact with the phone book. The menu options include adding, deleting, modifying, searching, displaying, and counting records, as well as quitting.

# Usage

To use the Phone Book System, compile and run the Phonebook2023.java file. The program will load any existing records from the 'records.txt file.

Users can interact with the phone book through the console menu by selecting various options.

The program will handle the requested operations and save the updated records back to the file after each modification. Users can exit the program by choosing the "Quit" option from the menu.

File Format records.txt)

The phone book records are stored in a comma-separated value (CSV) format in the records.txt file. Each line represents a single record with the following order of fields:

"first Name, last Name, address, city, phone Number".

Example of records.txt content:

Sadisu, Mohammed, zone 4, abuja, +2348126342742

Note: The file should be located in the same directory as the Phonebook2023.java file.

Limitations

• The program does not perform any data validation on user input. Users are expected to provide valid input.

Phone numbers are assumed to be unique identifiers for records.

Two records with the same phone number are considered duplicates.

• The program uses a simple CSV file to store records. For larger-scale applications, a more robust database or data storage mechanism should be considered.

# 

# Conclusion

The Phone Book System is a simple Java program that provides essential functionality for managing a phone book. It allows users to add, delete, modify, search, display, and count records.

The program is designed to be easy to use and serves as a basic demonstration of record management with file persistence.

For more complex applications, additional features and data validation may be required.