# *Telephone Directory*

PROJECT REPORT

By

**Aditya Mathur (RA2211003010096)**

**Om Arora (RA2211003010120)**

**Rahul Shankar Mathur (RA2211003010116)**

Under the guidance of   
**Dr. Kiruthika Devi S***In partial fulfilment for the Course*

of

**21CSC201J – Data Structures and Algorithm**

in <**C. TECH**>



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR**

**October 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that this minor project report for the course **21CSC201J** **DATA STRUCTURES AND ALGORITHM** entitled in "**Telephone Directory in Java**" is the Bonafide work of **Aditya Mathur (RA2211003010096), Om Arora (RA2211003010120) & Rahul Shankar Mathur (RA2211003010116)** who carried out the work under my supervision.

# SIGNATURE

<Dr.Kiruthika Devi S>

# <Designation>

**<Department>**

SRM Institute of Science and Technology

Kattankulathur

***Problem Statement***

# In the contemporary era of rapidly evolving telecommunications services, the imperative of efficient contact information management has never been more pronounced. Across various industries, particularly those with a dispersed workforce operating from different geographical locations, the telephone stands as the primary and often sole means of communication. In such diverse and dynamic settings, a seamless and user-friendly method for storing and swiftly retrieving contact details becomes an absolute necessity.

# The global workforce landscape has undergone a remarkable transformation in recent years, with businesses and organizations increasingly adopting remote work models, distributed teams, and geographically dispersed offices. This shift has underscored the significance of maintaining up-to-date contact information, as the ability to reach colleagues and associates promptly is pivotal to the smooth functioning of daily operations. Amidst these changes, the traditional rolodex or physical phone directory has given way to digital solutions that offer speed, accessibility, and versatility.

# With a growing reliance on telecommunications for professional communication, the demand for comprehensive contact management systems has surged. These systems not only store basic contact details like names and phone numbers but also often integrate additional information such as email addresses, departmental affiliations, or even role-specific details. Such enhanced functionalities not only facilitate effective communication but also enable businesses to optimize internal processes, boosting overall productivity and efficiency.

# In the pursuit of streamlining contact information management, companies and individuals have turned to diverse tools and applications. These can include contact management software, cloud-based databases, or mobile apps that enable synchronization across multiple devices. The ever-increasing array of technological solutions empowers users to not only store and search contact details but also to categorize, update, and protect this critical data.

# In conclusion, the current telecommunications landscape, marked by the prevalence of remote work and global communication, necessitates robust and user-friendly solutions for contact detail storage and retrieval. These solutions serve as a fundamental component in sustaining efficient and seamless professional interactions and relationships. With technology continuously advancing, the evolution of contact management tools promises to remain instrumental in fostering productivity and connectivity in our modern, dynamic world.

# 

# 

***Data Structure Concept used***

The provided code is an implementation of a basic phone book application with a graphical user interface (GUI). It employs various data structures and concepts to manage and manipulate contact data.

**1. Class "Contact":**

- Custom Data Structure: The `Contact` class represents a custom data structure that encapsulates information about a contact, consisting of a name, a phone number, and a reference to the next contact. It is essentially a linked list node as it contains a reference to the next `Contact`.

**2. Linked List:**

- `Contact` Instances as Linked List Nodes: The `Contact` instances form a linked list where each contact (node) is connected to the next one through the `next` reference. This linked list data structure allows the code to organize and manage a list of contacts efficiently, with each contact having a name, phone number, and a reference to the next contact.

**3. GUI Components:**

- JFrameThe main application window is represented by the `JFrame`, a Swing component that provides the GUI framework.

- JTextField: `nameField` and `phoneField` are used to input contact information (name and phone number).

- JButton: Various buttons (`addButton`, `searchButton`, `deleteButton`, `displayButton`) are used for user interaction.

- JTextArea: The `outputArea` is used for displaying output messages and contact information.

**4. Data Storage and Retrieval:**

- Contact List (Linked List): The list of contacts is maintained using a custom linked list data structure, where each `Contact` node is linked to the next one using the `next` reference.

- File I/O: The code uses file input/output (I/O) to save and load contact data. It reads and writes contact information to/from external files, allowing for data persistence.

**5. Event Handling:**

- Action Listeners: Action listeners are used to respond to user interactions with the GUI components. For instance, when the "Add Contact" button is clicked, an action listener is triggered to add a new contact.

**6. Error Handling:**

- Try-Catch Blocks: Try-catch blocks are used to handle exceptions, such as file-related errors when saving or loading contacts. This ensures that the program can gracefully handle unexpected issues.

**7. User Interface Customization:**

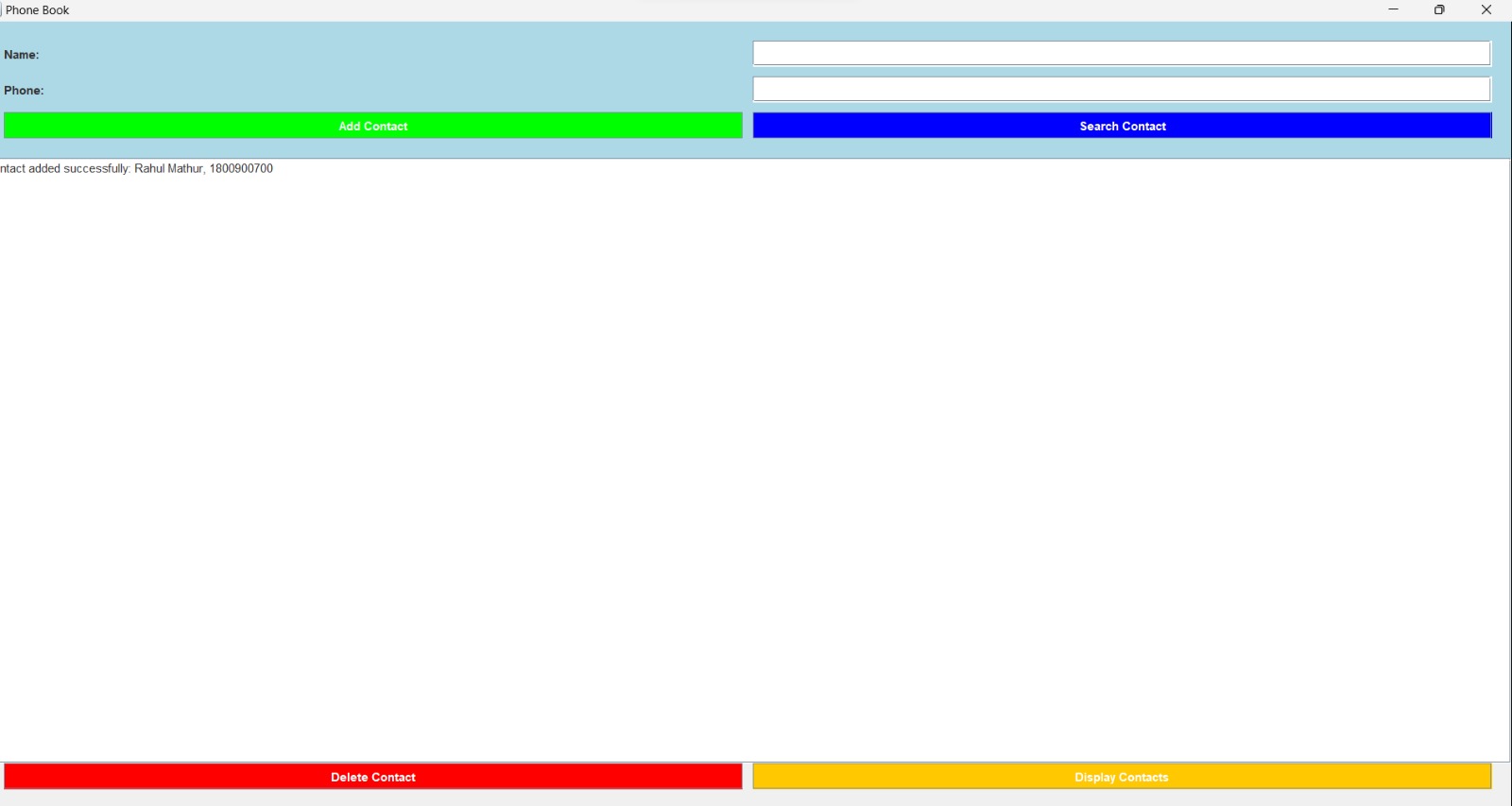
- Customization of GUI Components: Various methods (`customizeButton`, `customizeTextArea`, `customizeMenuItem`) are used to customize the appearance and behavior of GUI components, such as buttons, text areas, and menu items.

**8. Main Method:**

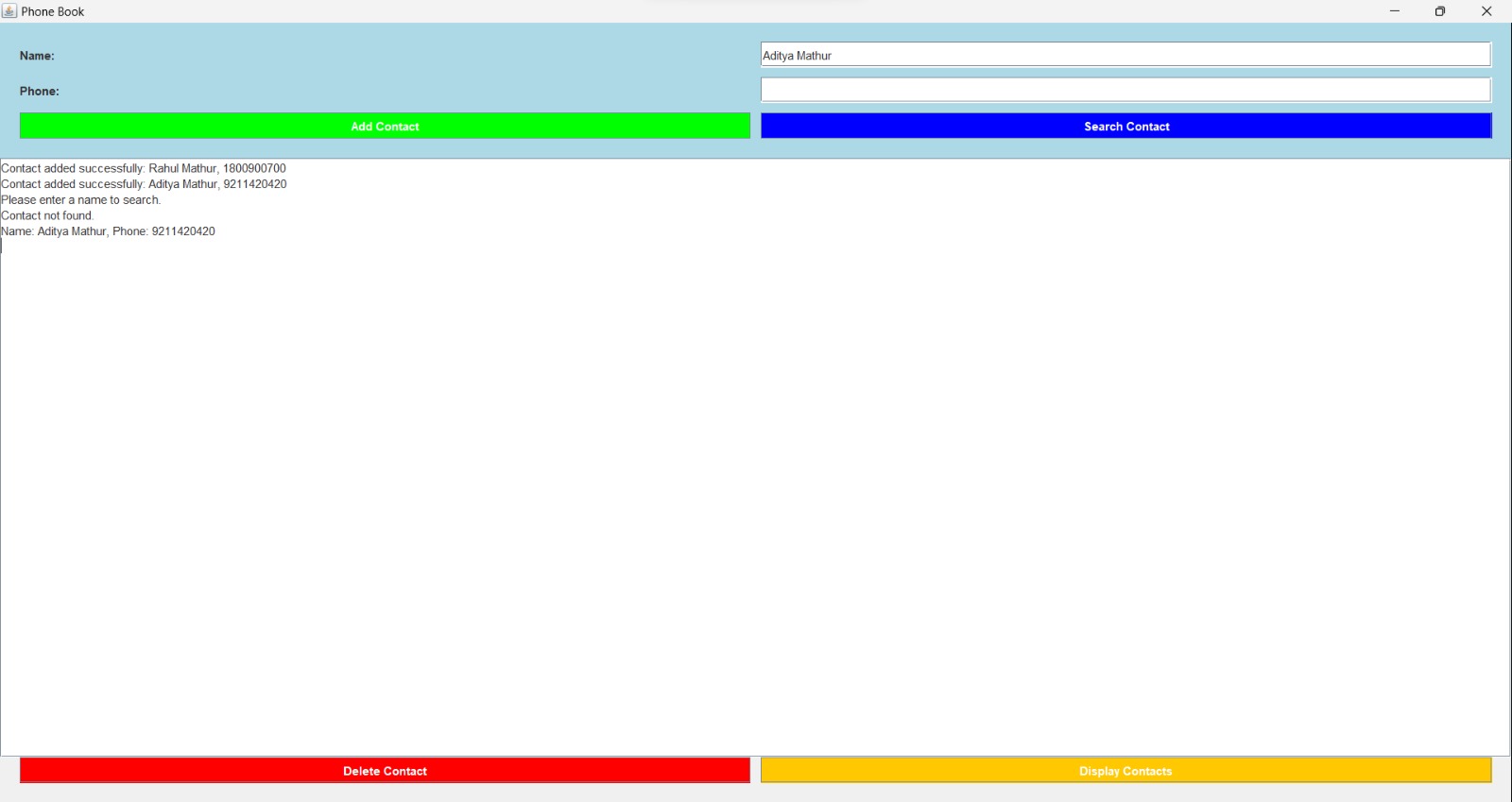
- Swing Thread: The `main` method utilizes SwingUtilities to execute the GUI-related code on the Swing event dispatch thread, ensuring proper synchronization of GUI operations.

***Demo output***

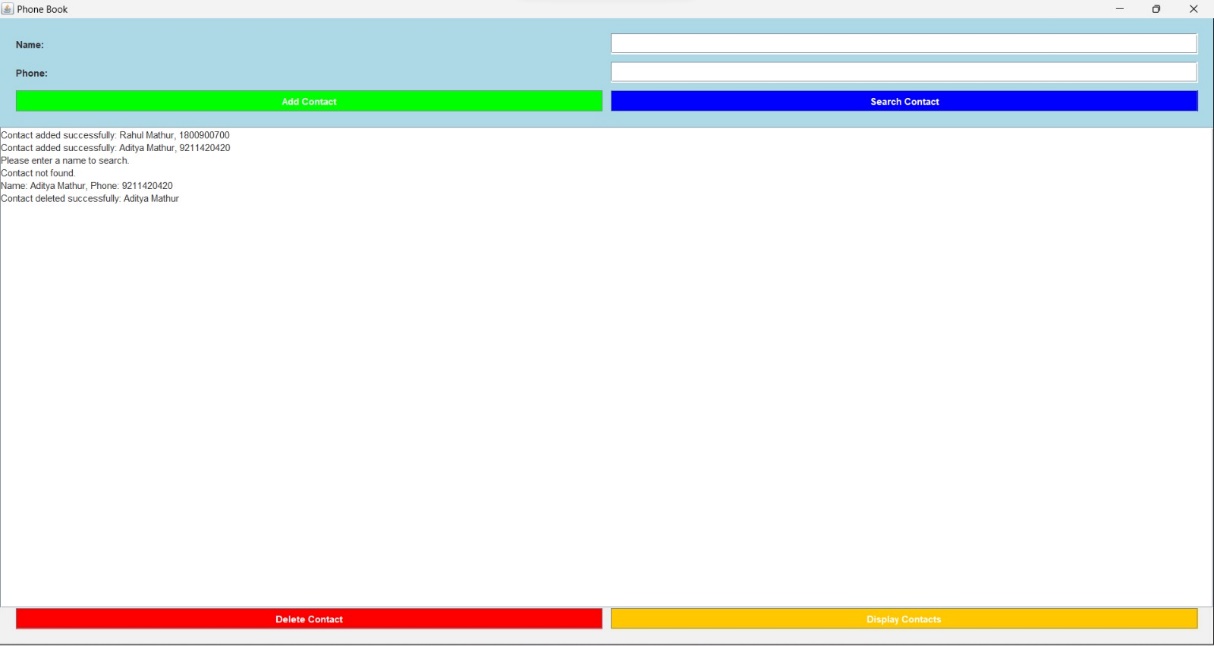
* **GUI:**



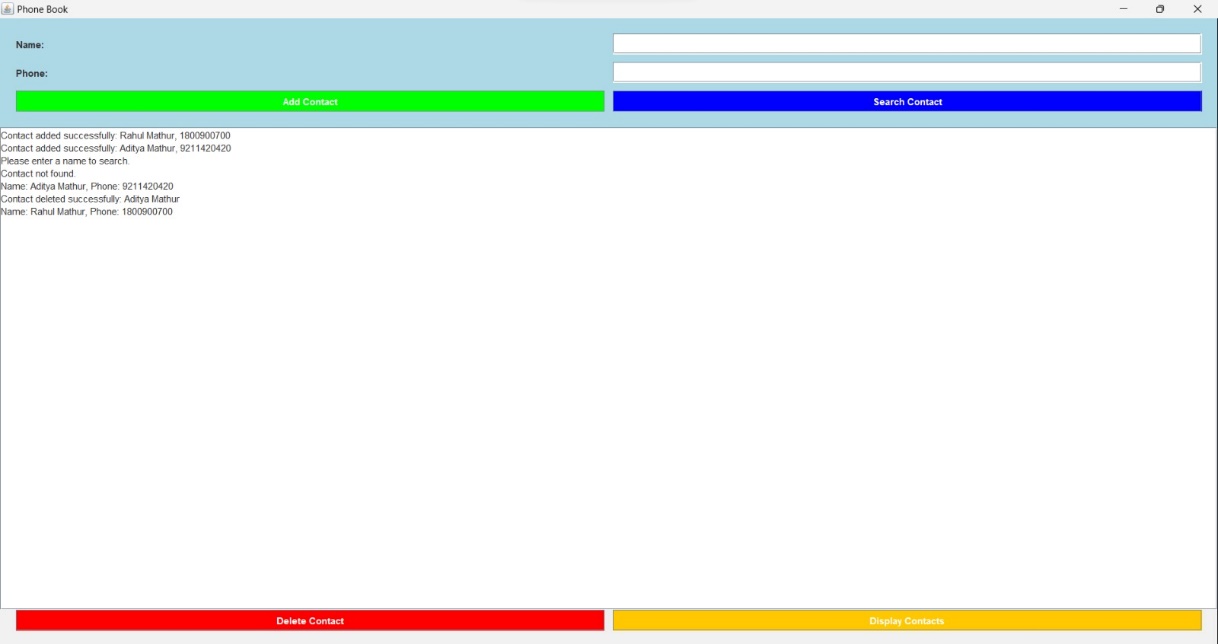
* **Using Search Feature:**



* **Using Delete Feature:**



* **Using Display Feature:**



***Conclusion***

In essence, the code elegantly combines a set of essential programming elements to craft a straightforward yet effective phone book application. Key among these is the strategic use of data structures, with linked lists serving as the backbone for managing contact information. This linked list structure allows for the efficient organization and retrieval of contact data, streamlining the process of adding, searching, and deleting entries.

Moreover, the code demonstrates a keen understanding of graphical user interface (GUI) design principles, seamlessly integrating GUI components to facilitate user interactions. These components, such as text fields and buttons, empower users to input, search, and manipulate contact details, thereby enhancing the user experience.

In addition to data structures and GUI components, the code exhibits proficiency in a range of programming concepts. Event handling mechanisms ensure that user actions trigger appropriate responses, guiding the flow of the application. Error handling techniques are skillfully implemented to gracefully manage unexpected issues, maintaining the stability and reliability of the program.

Furthermore, the code addresses the critical aspect of data storage and retrieval. By employing file input/output operations, it allows users to save and load contact information, granting the application data persistence. This not only safeguards the contacts but also enables users to seamlessly access their data across multiple sessions.

In summary, this code synthesizes data structures, GUI components, event handling, error management, and data storage to construct a user-friendly and functional phone book application. Its ability to efficiently manage contact information and provide a visually intuitive interface underscores its pragmatic approach to solving a common organizational challenge in the digital age.