

Phonebook Application Documentation

Table of Contents

1. [Project Overview](#)
 2. [Features](#)
 3. [Architecture](#)
 4. [Installation](#)
 5. [Usage](#)
 6. [Components](#)
 - o [Classes](#)
 - o [Methods](#)
 7. [User Interface](#)
 8. [Error Handling](#)
 9. [Future Enhancements](#)
 10. [Conclusion](#)
-

Project Overview

The Phonebook Application is a Java-based GUI application that allows users to manage their contacts efficiently. Built using the Swing framework, it provides features for adding, updating, deleting, and searching contacts. This application is designed to be user-friendly and visually appealing, making it easy for users to interact with their contact information.

Features

- **Add Contacts:** Users can add new contacts by entering a name and phone number.
- **Update Contacts:** Existing contacts can be updated with new information.
- **Delete Contacts:** Users can delete contacts from the phonebook.
- **Search Functionality:** Users can search for contacts by name or phone number.
- **Sorted View:** Contacts are automatically sorted by name for easy browsing.
- **User-Friendly Interface:** The application has a simple and visually appealing design with a color scheme to enhance usability.

Architecture

The application is structured into several components:

- **Model:** Represents the data (Contact).
- **View:** User interface components (JFrame, JTable, etc.).
- **Controller:** Action listeners that handle user interactions.

Installation

1. **Requirements:** Ensure you have Java Development Kit (JDK) installed on your machine.
2. **Clone the Repository:** Clone the project from the repository (if hosted).
3. **Compile and Run:** Use an IDE (like IntelliJ IDEA or Eclipse) or command line to compile and run the `PhonebookGUI` class.

Usage

1. **Launch the Application:** Run the `PhonebookGUI` main class.
2. **Add Contacts:** Fill in the name and phone number fields and click the "Add" button.
3. **Update Contacts:** Select a contact from the table, modify the fields, and click "Update."
4. **Delete Contacts:** Select a contact and click the "Delete" button.
5. **Search Contacts:** Type in the search field and click "Search" to filter the contacts displayed.

Components

Classes

- **Contact:** Represents a contact with a name and phone number.
- **PhonebookGUI:** The main class for the GUI, handling user interactions and displaying the contact list.

Methods

- `addContact()`: Adds a new contact to the list and table.
- `updateContact()`: Updates the selected contact's information.
- `deleteContact()`: Deletes the selected contact from the list and table.
- `searchContacts()`: Searches for contacts based on user input.
- `sortContacts()`: Sorts the contacts by name.
- `refreshTable()`: Refreshes the displayed contact table.
- `clearInputFields()`: Clears input fields after an operation.
- `showErrorMessage(String message)`: Displays error messages to the user.

User Interface

The application features a clean and organized user interface with:

- A **table** for displaying contacts.
- **Text fields** for inputting contact details and searching.
- **Buttons** for adding, updating, deleting, and searching contacts.

Color Scheme

- Background: Alice Blue
- Buttons: Steel Blue
- Text: Midnight Blue

Error Handling

The application includes error handling for various scenarios:

- **Input Validation:** Alerts users if they attempt to add or update contacts without entering required information.
- **Contact Selection:** Notifies users if they try to update or delete a contact without making a selection.

Future Enhancements

1. **Data Persistence:** Implement functionality to save and load contacts from a file or database.
2. **Contact Grouping:** Add the ability to group contacts into categories or tags.
3. **User Authentication:** Introduce user login functionality for personal contact management.
4. **Enhanced Search:** Implement advanced search options for filtering contacts by multiple criteria.

Conclusion

The Phonebook Application is a simple yet effective tool for managing contacts. With its intuitive interface and essential features, it serves as a solid foundation for further development and enhancement in contact management applications.