**PHONE CONTACT**

**A.DHARANI**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE.NO** |
| **1** | **AIM OF THE PROJECT** | **1** |
| **2** | **BUSINESS PROBLEM** | **1** |
| **3** | **PROJECT DESCRIPTION** | **2** |
| **4** | **FUNCTIONALITIES** | **3** |
| **5** | **INPUT VERSATILITY AND ERROR HANDLING** | **4** |
| **6** | **CODE IMPLEMENTATION** | **5** |
| **7** | **RESULT AND OUTCOMES** | **7** |
| **8** | **CONCLUSION** | **7** |

**PHONE CONTACT**

1. **­AIM OF THE PROJECT :**

Phone contact is a widely used kind of application. Probably we have seen earlier that our parents used to store information about our family members, friends, coworkers, and so on in a diary in the form of names and contact numbers. Contact lists are created and used for a myriad of reasons, and many people have different contact lists to meet their different needs. A contact book is used to store person’s contact like name, phone number etc. In this project, we'll build our own contact book using the python programming language.

1. **BUSINESS PROBLEM OR PROBLEM STATEMENT :**

In today's fast-paced business environment, maintaining accurate and up-to-date phone contact information is crucial for effective communication and relationship management. Currently, many organizations face challenges in managing their phone contacts due to inconsistencies, outdated information, and inefficient processes. The current phone contact management system is inefficient and prone to errors, leading to communication breakdowns, missed opportunities, and increased operational costs. Specifically, the problems include :

**2.1 Duplicate Contacts**: Multiple entries for the same contact, leading to confusion and redundancy.Outdated Information: Contacts often have outdated phone numbers or missing critical details.

**2.2 Inefficient Updates**: Manual updating processes are time-consuming and prone to human error.

**2.3 Lack of Integration**: The contact management system is not well integrated with other business systems (e.g., CRM, email), leading to soiled data and reduced productivity.To develop and implement a streamlined and integrated phone contact management system that,Eliminates duplicate entries and ensures each contact is unique. Keeps contact information up-to-date through automated updates and validation processes. Reduces the time and effort required to manage contacts through user-friendly interfaces and automation. Integrates seamlessly with other business systems to provide a holistic view of contact information across the organization. Expected Benefits:

* Improved accuracy and reliability of contact information.
* Enhanced communication efficiency and effectiveness.
* Reduced operational costs related to manual data entry and correction.
* Better alignment and integration with other business systems, leading to improved overall productivity.
* By addressing these issues, the organization aims to enhance its communication infrastructure, support better decision-making, and ultimately improve customer satisfaction and business outcomes.

**3. PROJECT DESCRIPTION:**

The Phone Contact Management System is a Python-based application that provides users with a simple and efficient way to manage their phone contacts. The system will allow users to add, search, display, edit and delete, exit contact information. Each contact will contain details such as name, phone number. Features,

* Add contact
* Search contact
* Display contact
* Edit contact
* Delete contact
* Exit contact

**METHODOLOGIES:**

* Object-Oriented Programming (OOP):
* Exception Handling:
* Command-Line Interface (CLI):

**4. FUNCTIONALITIES :**

**Functionalities provided by Contact Management System are as follows:**

Creating a phone contact management system in Python involves several functionalities, such as adding, searching, display, edit, deleting, and edit contacts. Provides the searching facilities based on various factors. Such as name,Contact. Manage the information of Credential. Shows the information and description of the Contact. To increase efficiency of managing the Contact, Credential. Manage the information of phone Contact. Editing, adding, searching , displaying, exiting and deleting of Records is improved which results in proper resource management of phone Contact . Below is an example of how you might implement such a system using a simple text-based menu interface and storing contacts in a list of dictionaries.

**Here’s a basic implementation of these functionalities:**

**4.1 ADD CONTACT :**

When the user selects the option to add a new contact, the script prompts them to enter the contact's name and phone number. These inputs are then stored in the contact dictionary, where the name serves as the key and the phone number as the value. After successfully adding the contact, a confirmation message is displayed.

**4.2 SEARCH CONTACT:**

This option allows the user to search for a contact by name. The script prompts the user to enter the name they wish to search for. It then checks if the name exists in the contact dictionary. If found, the corresponding phone number is displayed. If not, a message indicating that the contact was not found is shown.

**4.3 DISPLAY CONTACT:**

When the user opts to display all contacts, the script checks if the contact dictionary is empty. If the dictionary is empty, it informs the user that the contact book is empty. Otherwise, it displays all contacts in a formatted manner, showing the name and corresponding phone number in a tabular format.

**4.4 EDIT CONTACT :**

To edit a contact, the user is prompted to enter the name of the contact they wish to edit. The script checks if the name exists in the contact dictionary. If it does, the user is prompted to enter a new phone number for the contact. The contact's phone number is then updated with the new value, and a confirmation message is displayed. If the contact name is not found, the script informs the user accordingly**.**

**4.5 DELETE CONTACT** :

For deleting a contact, the user is prompted to enter the name of the contact they wish to delete. The script checks if the contact exists in the contact dictionary. If found, it asks the user to confirm the deletion by typing 'y' or 'Y'. If the user confirms, the contact is removed from the dictionary, and a confirmation message is displayed. If the user cancels or the contact is not found, appropriate messages are shown .

**4.6 EXIT CONTACT:**

The user can exit the contact management application by selecting the exit option. When this option is chosen, a farewell message is displayed, and the loop terminates, ending the script.

1. **INPUT VERSATILITY WITH ERROR HANDLING AN EXCEPTION HANDLING :**

The provided contact management script is a basic command-line application that allows users to add, search, display, edit, and delete contacts. The code is designed to handle various user inputs and potential errors gracefully, ensuring a smooth user experience.

**Main Menu Selection Error Handling:**

This block handles the input for the main menu choice. int(input(...)) attempts to convert the user's input to an integer. If the input is not a valid integer, a ValueError is raised. The except ValueError: block catches this error and prints an error message, then uses continue to restart the loop, allowing the user to try again.

**Adding a New Contact Error Handling:**

This block handles the input for adding a new contact. The code attempts to read the contact name and phone number from the user. int(input(...)) converts the phone number to an integer. If any error occurs (e.g., ValueError if the phone number is not a valid integer), the except Exception as e: block catches it. The error message is printed, indicating what went wrong.

**Editing a Contact Error Handling:**

This block handles the input for editing an existing contact. The code checks if the contact exists in the dictionary. If the contact exists, it attempts to read a new phone number and convert it to an integer. If any error occurs during this process, the except Exception as e: block catches it and prints an error message.

**6. CODE IMPLEMENTATION :**

Creating a phone contact project in Python involves building a program that allows you to add, view, search, update, and delete, exit contacts. Below is a simple implementation of such a project using Python. This program uses a dictionary to store the contacts, with names as keys and phone numbers as values.

**IMPLEMENTATION :**

* Initializes an empty dictionary to store contacts.
* Add­­­\_contact: Adds a new contact if the name doesn't already exist.
* view\_contacts: Displays all contacts.
* search\_contact: Searches for a contact by name.
* update\_contact: Updates the phone number of an existing contact.
* delete\_contact: Deletes a contact by name.

**syntax :**

**7. RESULT AND OUTCOME :**

Below is a hypothetical session showing how a user might interact with the Phone Contact Manager.

**7.1 ADDING CONTACTS:**

The user adds two contacts, Alice and Bob, with their respective phone numbers.The program confirms the addition of each contact.

**7.2 VIEWING CONTACTS:**

The user views all contacts, and the program lists Alice and Bob with their phone numbers.

**7.3 SEARCHING FOR A CONTACT:**

The user searches for Alice, and the program displays Alice's phone number.

**7.4 UPDATING A CONTACT:**

The user updates Alice's phone number. The program confirms the update.The user views the contacts again to verify the update.

**7.5 DELETING A CONTACT:**

The user deletes Bob's contact.The program confirms the deletion.The user views the contacts again to verify Bob's contact has been removed.

7.6 EXITING THE PROGRAM:

The user chooses to exit, and the program terminates.

**8. CONCLUSION :**

The phone contact project in Python demonstrates the creation of a simple, yet functional, contact management system. This project allows users to add, view, search, update, and delete contacts through an intuitive, menu-driven interface. The implementation leverages Python's dictionary data structure to store and manage contacts, ensuring efficient and straightforward access to the contact information**.**