Milan Thapa, Arshul Shaik, Miriam Luka, Hayrenzy Hassen

INST 326

05/08/23

## Documentation

Along with your code, unit tests, and any supporting files, you will also need to provide some documentation. This documentation should contain:

**1 . One to four sentences explaining what your project is and what it does.**

The proposed project is a contact management application that will allow the user to keep track of numerous contacts through a user interface. The application will feature a contact list, search functionality, categorization of communications, and the ability to add/remove contacts. The benefits of the application include improved organization and time-saving for managing and communicating with contacts.

**2 . An explanation of how to run the program from the command line.**

This program is a simple command-line interface for managing contacts in an SQLite database. It provides several options for adding, editing, deleting, searching, and grouping contacts by category.

**3. Documentation on how to use the program / how to interpret the output of the program.**

Getting Started

To get started with the program, you need to have Python 3 installed on your computer, as well as the sqlite3 module, which comes pre-installed with Python.

The first step is to download the program files and save them to a directory on your computer. Then, open a terminal or command prompt and navigate to the directory where you saved the files.

Usage

To run the program, simply type python contact\_manager.py in the terminal and press Enter. This will start the program and display a menu with several options:

1. Add a new contact

2. Edit an existing contact

3. Delete a contact

4. Search for contacts

5. Group contacts by category

6. Exit

You can select an option by entering its number and pressing Enter.

Adding a New Contact

To add a new contact, select option 1 from the menu. Then, follow the prompts to enter the contact's first name, last name, email, phone number, and category.

Editing an Existing Contact

To edit an existing contact, select option 2 from the menu. Then, follow the prompts to enter the ID of the contact you want to edit and the new values for the contact's first name, last name, email, phone number, and/or category.

Deleting a Contact

To delete a contact, select option 3 from the menu. Then, follow the prompts to enter the ID of the contact you want to delete.

Searching for Contacts

To search for contacts, select option 4 from the menu. Then, follow the prompts to enter a search term. The program will search for contacts whose first name, last name, email, phone number, or category contains the search term, and display a list of matching contacts.

Grouping Contacts by Category

To group contacts by category, select option 5 from the menu. The program will display a list of categories and the number of contacts in each category.

Exiting the Program

To exit the program, select option 6 from the menu.

Output

The program provides various types of output, depending on the selected option:

When adding a new contact, the program displays the ID of the newly created contact.

When editing or deleting a contact, the program displays a message indicating whether the operation was successful or not.

When searching for contacts, the program displays a list of contacts that match the search term, including their ID, first name, last name, email, phone number, and category.

When grouping contacts by category, the program displays a list of categories and the number of contacts in each category.

Conclusion

This program provides a simple way to manage contacts in an SQLite database. With just a few commands, you can add, edit, delete, search, and group contacts by category.

**4. An annotated bibliography of all sources you used to develop the program and how you used them.**

Python Software Foundation. (2021). sqlite3 - DB-API 2.0 interface for SQLite databases. Retrieved May 8, 2023, from https://docs.python.org/3/library/sqlite3.html

Python's sqlite3 module documentation:

The official documentation for the sqlite3 module in Python provides detailed information on how to use it for SQLite database operations. It covers topics like creating a connection, creating tables, executing SQL statements, committing transactions, and more. It was used to understand the syntax and usage of the sqlite3 module functions used in the program.