

Phone Book Management App

Application Structure

The app contains five Python files:

- main.py - serves as the entry point
- contact.py - holds the Contact structure and a few internal methods
- phonebook.py - contains all Phone Book Management functions
- custom_logger.py - handles logging and auditing
- utils.py - consists of utility functions for validation

Modules and Packages

The application uses three main packages for validation, display, and search functions.

- email-validator
- prettytable
- fuzzywuzzy

Data Structure

The Contact class contains properties and a list of internal functions.

Contact properties:

- first_name
- last_name
- phone
- email
- address
- __created_date
- __is_deleted

Contact functions:

- __init__(dict, bool, bool)
- created_date()
- delete()
- is_deleted()
- __str__()

Functionalities

The app includes basic phone book management functionalities such as:

- Create (command: C)
- Update (command: U)
- Query (command: Q)
- Delete (command: D)
- List (commands: L, LA or LD)
- Filter (command: LF)
- Upload (command: F)

Create

Contact creation requires three properties: `first_name`, `last_name`, and `phone`. Email and address are optional. Phone and email are validated, with phone following the pattern (xxx) xxx-xxxx (note the space between ") and "x").

Update

Update operations use a unique Contact UUID. Providing the UUID initiates a process similar to contact creation. It also allows users to skip unchanged fields.

Query

With the "fuzzywuzzy" package, users can search for contacts using any available information - first name, last name, phone number, or a combination.

Delete

Deletion requires a valid contact UUID, and the user must confirm their decision by typing "Y".

List

Contacts are stored in the order they are created. The basic "L" command lists them by default, while "LA" lists them in ascending order and "LD" in descending order by last name.

Filter

Filtering is a specific listing (L) within a given time frame. By providing a start and end date, you can list contacts accordingly.

Upload

The app allows uploading contacts from a CSV file. By providing the complete CSV file path, all contacts are loaded into the phone book. Additionally, the upload supports the `created_date` field, which is protected in manual contact creation.