# Report: Phone Book Manager Application

My repo link: https://github.com/jiayiwei9/GNG5300

## Overview

The Phone Book Manager is a Python program that helps users manage their contacts. You can easily add, search for, view, update, and delete contacts. This report explains how the app works, what tools we used, the features included, and the progress made during its development.

Overview of the Application Files:

contact.py: This file defines the Contact class, which represents individual contact entries. It includes methods for validating phone numbers and email addresses, formatting contact information, and maintaining a history of changes made to each contact.

phonebook.py: This file contains the PhoneBook class, which manages all contact operations. It provides methods for adding, viewing, searching, updating, and deleting contacts. It also supports importing contacts from CSV files and includes functionality for sorting and grouping contacts.

utils.py: This file includes utility functions for validating phone numbers and email addresses. These functions ensure that user input meets the required formats before contacts are created or updated.

main.py: This is the entry point of the application. It sets up a command-line interface that allows users to interact with the PhoneBook class. Users can choose various options to manage their contacts through a simple menu.

## Application Structure

The application has a few main parts:

**Contact Class:**

This represents a contact with details like first name, last name, phone number, email, address, and timestamps for when the contact was created and last updated.

1.validate\_and\_format\_phone\_number(): Checks if the phone number is in the right format.

2.validate\_email(): Makes sure the email address is valid.

.update\_history(): Keeps track of changes made to the contact.

**PhoneBook Class:**

This manages a list of contacts and has methods for different operations.

1.add\_contact(): Adds a new contact to the list.

2.view\_contacts(): Displays all contacts.

3.search\_contacts(): Finds contacts by name or phone number.

4.update\_contact(): Changes details of an existing contact.

5.delete\_contact(): Removes a contact from the list.

6.import\_contacts\_from\_csv(): Imports contacts from a CSV file with flexible header options.

7.sort\_contacts(): Sorts contacts by first or last name.

8.group\_contacts\_by\_initial(): Groups contacts by the first letter of their names.

9.filter\_contacts\_by\_date(): Finds contacts created within a specific date range.

**Utilities Module:**

Contains functions for validating phone numbers and email addresses.

1.validate\_phone\_number(): Checks if the phone number format is correct.

2.validate\_email(): Validates the email format.

**Main Script**

This is the part of the program where users interact with the PhoneBook class. It provides a menu for users to choose what they want to do.

## Data Structures Used

**List**: We use a list to store multiple Contact objects, making it easy to add or remove contacts.

**Dictionaries**: These are used in the CSV import to match different header names to the expected contact details, making it easier for users.

**Strings and Timestamps**: Each Contact uses strings for names and date/time formats for creation and updates.

## Development Progress

The Phone Book Manager has gone through several important improvements:

**Initial Version**: The first version allowed basic tasks like adding and deleting contacts.

**Fixes and Improvements**:

**1.Contact History Bug**: We fixed a problem that happened when trying to view a contact’s history.

**2.Mandatory Fields**: We made sure that first and last names are required when adding or updating contacts.

**3.CSV Import Flexibility**: The CSV import was changed to accept similar headers instead of needing exact matches, which makes it easier to use.

**Better Error Handling**: We improved feedback for users when they made mistakes, like selecting the wrong contact index for updates or deletions.

**User Feedback**: We made changes based on what users said to improve the overall experience.

## Features Included

1. **CRUD Operations**: Users can create, read, update, and delete contacts.
2. **Advanced Search**: Search for contacts using partial names or phone numbers.
3. **CSV Import**: Easily import multiple contacts from a CSV file.
4. **Sorting and Grouping**: Sort contacts by name and group them by the first letter.
5. **History Logging**: Keep a record of changes made to each contact.
6. **Input Validation**: Check that phone numbers and emails are in the right formats to avoid errors.

## Conclusion

The Phone Book Manager has become a handy tool for managing contacts. It combines a simple command-line interface with useful features. The design and improvements show a focus on quality and user satisfaction. Overall, this application is a great example of using Python for practical tasks like contact management.