

## COMP1002 – Mini-Project Report Template

**Project Title:**

Project 2: Contact Management System

**Team Members:**

23088706D Chat Kiu Yin  
23088621D Cheung Kei Yau  
23078028D Chong Zhi Yi  
23082545D Lee Yin Lam Elaine

**Course Title:** Computational Thinking and Problem Solving

**Instructor:** Dr. Muhammad Tayyab

**Department:** Computing

**Institution:** The Hong Kong Polytechnic University

**Submission Date:** 27/11/2023

## **Table of Contents**

**(use the MS word auto generate feature to get table of contents)**

• COMP1002 – Mini-Project Report Template.....	1
• Introduction .....	3
• Objectives.....	3
• Project Design .....	4
• Implementation Details.....	7
• Libraries and Tools Used .....	8
• Testing and Results.....	8
• Conclusions .....	9
• References.....	10
• Note to Students:.....	12

## Introduction

Write an overview of the project:

- Briefly describe the project you were assigned.  
Our project is a contact management system which allows users to manage their contacts efficiently with various features. Such features include basic management of contacts, duplicate finding, contact categorization and fuzzy searching.
- Outline what the completed project aims to accomplish.  
Contact management, duplicate finding, categorization, fuzzy searching
- Discuss the intended audience or users of your project.  
Individuals or organizations that need effective management of their contacts. These could include professionals and businesses that need to deal with a substantial amount of contact information.

## Objectives

Detail what you aim to achieve:

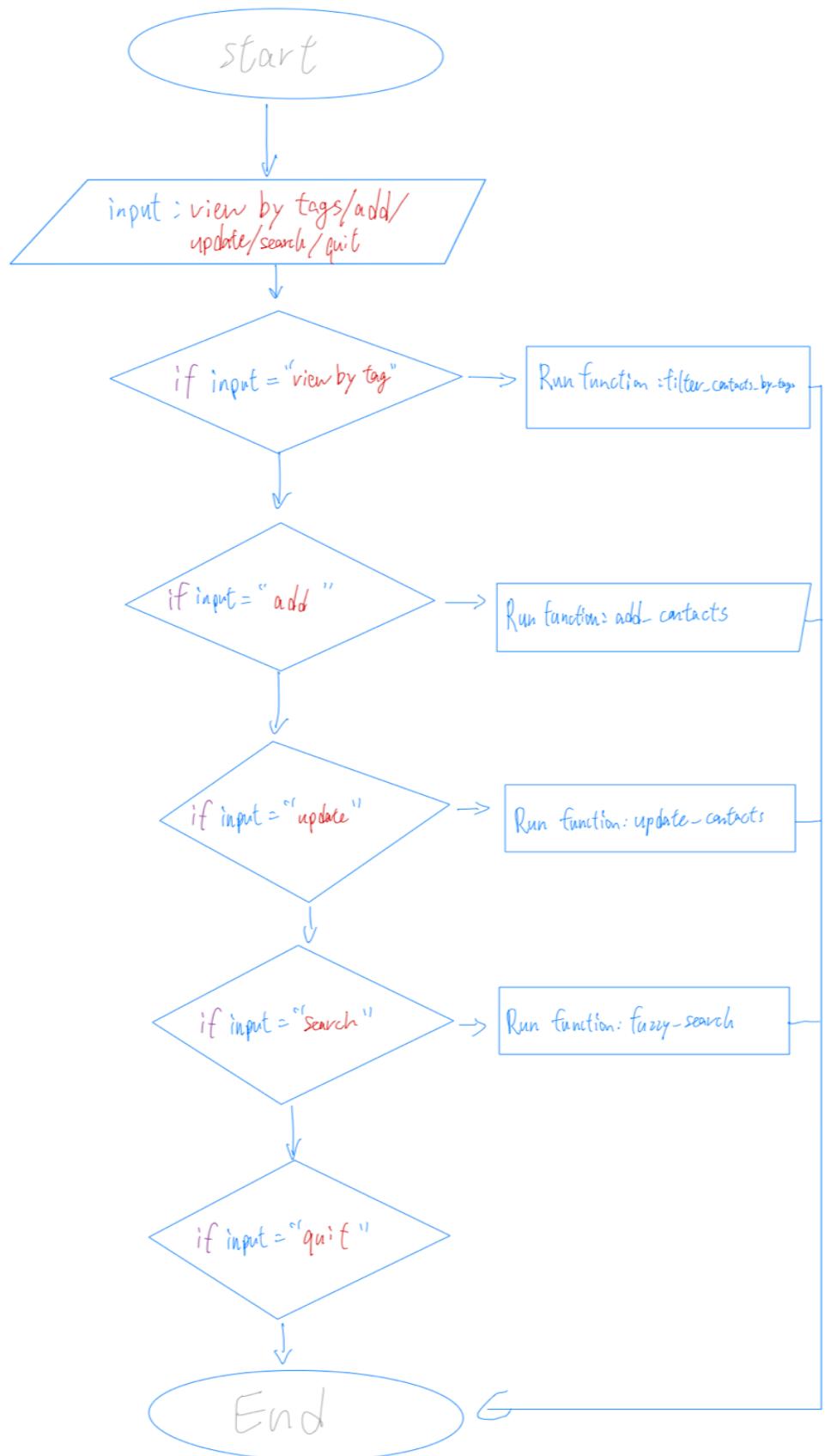
- Divide the project requirements in at most 5 objectives.  
Contact management  
Duplicate finding  
Categorization  
Fuzzy searching
- Outline the specific objectives as given with the project assignment.
  1. Contact management: Allows users to store, retrieve, update and categorize contact information. This includes designing a file structure, implementing functions to read/write from file, and providing a user-friendly interface for management.
  2. Duplicate finding: Allows users to identify duplicate contacts based on telephone number or name to remove duplicates and ensure data integrity.
  3. Categorization: Allows users to categorize contacts based on user-defined attributes, such as relationship type, organization or location.
  4. Fuzzy searching: Allows users to perform fuzzy searching on approximate matches on regex library.
- Explain how you plan to meet these objectives within the given timeframe.  
We defined the objectives that we are clear about what we are to achieve.

## Project Design

Describe your approach:

- This part only describes the project design idea, and not the actual implementation
- Explain the structure and design of your project, including its major components or modules.
- Include diagrams such as user interface mockups, class hierarchy diagrams, or software architecture diagrams to illustrate your design

The code starts by asking whether the user wants to perform the function of “view by tags” or “add” or “update” or “search” or “delete” or “quit”. Then, each function performs their respective functions. “View by tags” allows users to filter the contacts according to their tags. “Add” allows users to add a new contact. “Update” allows users to update their current contacts. “Search” allows users to perform fuzzy searching. “Delete” all users to delete a selected contact. “Quit” allows users to quit the running program.





## Implementation Details

Walk through the development process:

- Discuss the choice of programming constructs and control structures used in your code.

We used loops, conditionals and string formatting in our codes. We used for loop to iterate over lists (e.g., the list of contacts or the list of tags of a contact). We used while loop to keep the program running before the user quit it. We used if statement to perform actions based on certain conditions, such as whether a contact matches a search term or a tag, or whether the user chooses to add contacts, update contacts or filter contacts by tag, search contacts, or quit the program.

- Describe the major functions, classes, and modules, including their roles in the project.

Load\_contacts, save\_contacts, add\_contact, update\_contact, filter\_contacts\_by\_tag, fuzzy\_search, print\_contacts, delete\_contact.

Load\_contacts: Read the contacts from the file;

Save\_contacts: Save the contacts into the system;

Add\_contact: Add contacts to the system;

Update\_contact: Update the current contact;

Filter\_contacts\_by\_tag: Filter contacts by their tags;

Fuzzy\_search: Allows users to perform fuzzy searching on approximate matches on regex library;

Print\_contacts: Output the current contacts

delete\_contact: delete the selected contact.

- Mention any challenges faced during coding and how you approached solving them.

Debugging and improving the code was a challenge for us as the code is quite complicated. We encounter a challenge that we are not able to show the relationships (e.g. family, friends, colleagues) between the contacts. We can only show one of the relationships but are unable to show all of the relationships. The fuzzy search was unable to just look up a specific contact but in return printing out all contacts. Through trial and error, we successfully developed a feasible solution. We have difficulties on merging contacts and we are unable to finish the merge part.

- Mention something you optimized by yourself that require reader attention

Fuzzy search, it was modified to print according to format instead of printing each result of search result list.

## Libraries and Tools Used

- List external Python libraries and tools used in your project.

Json [1] (the contact file) and regex

- Briefly justify their usage and relevance to the tasks at hand.

Json is used for making up a list of contacts. Regex is used in fuzzy searching and updating contacts.

## Testing and Results

Define how you tested your project:

- Explain your testing approach and the types of testing you performed (unit tests, integration tests, user tests, etc.), you do not need the industry standard tests at this stage however overall mention the testing approach.

We used unit testing in fuzzy searching, by moving a module into a separated file.

- Describe any bugs or problems encountered and discuss the solutions implemented.

We encounter a challenge that we are not able to search with more than one tag. It returns nothing if we input more than one tag. The fuzzy search was unable to just look up a specific contact but in return printing out all contacts. Through trial and error, we successfully developed a feasible solution. We have a challenge that we unable to apply approximate match in all functions.

- Provide a summary of the test results and any user (someone outside your project group) feedback, if available.

```
Choose the mode you want (view by tags/add/update/search/quit): & C:/Users/aiden/AppData/Local/Programs/Python/Python311/python.exe c:/Users/aiden/OneDrive/Documents/VSCode/contactmgr/main.py
Invalid choice.
Enter the category to search in (name, phone, email, tags): name
Enter the search term: Jen
Merge(m) or remove (r)? r
Contact not found.
Enter the category to search in (name, phone, email, tags): name
Enter the search term: Jennifer Lopez
Contact not found.
Choose the mode you want (view by tags/add/update/search/quit): quit
PS C:\Users\aiden\OneDrive\Documents\VSCode\contactmgr> & C:/Users/aiden/AppData/Local/Programs/Python/Python311/python.exe c:/Users/aiden/OneDrive/Documents/VSCode/contactmgr/main.py
Choose the mode you want (view by tags/add/update/search/quit): update
Enter the category to search in (name, phone, email, tags): Jennifer Lopez
Enter the search term: none
Contact not found.
Enter the category to search in (name, phone, email, tags): name
Enter the search term: Jennifer Lopez
Merge(m) or Remove(r)? r
Enter the new phone number (leave blank to keep it unchanged):
Enter the new email address (leave blank to keep it unchanged):
Enter the new tags (comma-separated, leave blank to keep them unchanged):
Contact updated successfully.
Updated Contact:
Name: Jennifer Lopez
Phone: +1 (985) 150-2216
Email: jlopez@gmail.com
```

## Conclusions

Draw conclusions from your experience:

- Reflect on whether the project achieved the set objectives.
- Discuss what you have learned in terms of computational thinking and Python programming.
- Offer suggestions for improvements or additional functionality.
- Add your thoughts on future of this project and if it is possible to convert this into a real product

Our project achieved some of the set objectives. The objectives achieved are contact management and fuzzy search. After the project, we learnt the usage of functions. We defined a lot of functions in our coding project, such as load\_contacts, save\_contacts, add\_contacts, update\_contacts, filter\_contacts\_by\_tag, search\_contacts\_by\_name, print\_contacts, delete\_contact. We use these functions to provide the functions that the user needs to use. The code is quite complex and user-unfriendly, we should improve the code by simplifying the code and make it more user-friendly. If we were able to improve the system in the future, we would make it more user-friendly and much simpler.

## **References**

List all the resources used during the development, following a consistent citation style. APA is one option to do citation, however you can follow whichever you find easy to work with.

[1] Poe, 2023. ChatGPT (3.5) [for generating contact.json] [online]. Available:

<https://poe.com/s/90vWFXPNn6uvlqO0vnEM> [Accessed Nov 19, 2023].

## Appendices

- **A - User Manual:** Step-by-step instructions on how to use/ interact with the project.
  1. Run the script. It will load the contacts from the contacts.json file and print them.
  2. Choose an operation: view by tags, add, update, search, or quit.
  3. If you choose “view by tags”, you will be asked to input a tag. The script will then display all contacts that have the tag.
  4. If you choose “add”, you will be asked to input the name, phone number, email address, and tags (comma-separated) for the new contact. The new contact will be added to the contacts list.
  5. If you choose “update”, you will be asked to input the name of the contact to update. The updated contact will be added to the contacts list.
  6. If you choose “search”, you will be asked to input a search term. The script will then display all contacts that match this search term.
  7. If you choose “delete”, you will be asked to input a search term. The chosen contact will be deleted to the contacts list.
  8. If you choose “quit”, the script will save the contacts to the contacts.json file and exit.

9. If you input anything else, the script will print "Invalid choice" and ask you to choose an operation again.

All changes you made to the contacts are saved to the contacts.json file.

- **B - Code Listings:** Important code snippets or a link to where the full code can be found (e.g., a GitHub repository). Making the code closed or open source is up to you.
- <https://github.com/aidenchat/contactmgr.git>
- **C - Testing Evidence:** Screenshots of progress, tests, output results, or tables documenting test cases.

## Note to Students:

- **Language:** Maintain a formal but accessible tone appropriate for a technical document.
- **Consistency:** Adopt a uniform style for headings, subheadings, and fonts throughout the report.
- **Proofreading:** Review the report for clarity and check for typographical errors before submission.
- **Submission Format:** Submit the report in both PDF and DOCX format.
- **Backup:** Always have a backup of your report files and any related project files.