

# User Manual

Members: Huang Tianji (21099573d)<sup>1</sup>, Zhang Zhiyuan (21096414d)<sup>1</sup>,  
Zheng Shouwei (21097982d)<sup>1</sup>, Zhou Taiqi (21106717d)<sup>1</sup>  
Team: 20 1. Hong Kong Polytechnic University

## Table of Contents

<b>1. Functionalities of PIRs .....</b>	<b>1</b>
<b>2. Usage tutorial .....</b>	<b>2</b>
<b>2.1 Contact, Event, Task .....</b>	<b>2</b>
<b>2.2 QuickNote.....</b>	<b>4</b>
<b>2.3 Search.....</b>	<b>5</b>
<b>2.4 Load .....</b>	<b>5</b>
<b>2.5 Alarm.....</b>	<b>6</b>
<b>3. Precautions.....</b>	<b>6</b>
<b>4. PIRs Input Restrictions .....</b>	<b>6</b>
<b>5. Input Commands and Structure .....</b>	<b>6</b>
<b>6. Invalid Input Handling .....</b>	<b>6</b>
<b>7. Output Interpretation .....</b>	<b>7</b>
<b>8. Advanced Features .....</b>	<b>7</b>

## 1. Functionalities of PIRs

**Contacts:** Stores essential contact details such as **names**, **phone numbers**, **emails**, and **addresses**.

**Events:** Manages event-related information, including **descriptions**, **start times**, and customizable **alarm settings**. For events, we have set up a series of alarm mechanisms. In addition to returning through the command line within the program. If the program notices that some of the user's event times meet specific requirements during operation, it will automatically send an **email** prompting the event to the user's registered mailbox.

**Tasks:** Organize tasks, complete with **descriptive elements** and **deadlines**.

**QuickNotes:** Facilitates the creation of brief, text-based **notes**.

## 2. Usage tutorial

Users need to start the program from *main.py*. The command line will then display an interactive option. **Select** the type of instance to be created according to the type required by the user, or directly **load files**, **search** and **exit** directly.

```
[Running] python -u "/Users/arctic_zhou/Documents/GitHub/COMP3211_G
Personal Information Manager (PIM)
1. Contacts
2. Events
3. Take Quick Notes
4. Tasks
5. Load File
6. Search
9. Exit
Enter your choice (1-4, 9):
```

After selecting any instance type (**Enter 1,2,3,4**), the system will prompt the user whether to enter the name of the file to store the instance created this time. If 'y' is selected, the user will be asked to enter the name of the required file, and then a new file with this name will be created in PIM\_dbs; if 'n' is selected, a file of the corresponding type will be created in a default format. The file name contains the **current date**.

```
Would you like to self define the contact filename? (y/n) > 
```

After selecting the file to load('y'), the user will be prompted to enter the name of the file to be loaded. After verifying that the file is correct, the operation interface for the file is also entered.

```
Enter filename start with 'Contacts' > test
!!!!!!Not start with 'Contacts' try again!!!!!!
```

```
Enter filename start with 'Events' > Events_1
```

```
Enter filename start with 'Tasks' > Tasks_1
```

```
Enter filename start with 'QuickNotes' > QuickNotes_1
=====Take Quick Notes=====
=====Enter END to END=====
```

### 2.1 Contact, Event, Task

#### 1. Choice interface

After creating an instance storage file or loading a file, the user will be asked to choose what to do with the file. Operations are divided into **add**, **view** and **delete**. The function of **add** is to create the corresponding instance in the file. After selection, the user will be prompted to enter several attributes of the instance; the function of **view** is to return the instance that the user wants to query. After selection, the user will be prompted to enter the name of the instance; **delete** is used to delete An instance also requires the user to enter the corresponding name after selection.

(Contact choice interface)

```
Enter filename start with 'Contacts' > Contacts_test
1. Add Contact
2. View Contacts
3. Delete Contact
4. Back to Main Menu
Enter your choice (1-4): 
```

(Event choice interface)

```
1. Add Event
2. View Events
3. Delete Event
4. Back to Main Menu
```

(Task choice interface)

```
1. Add Task
2. View Tasks
3. Delete Task
4. Back to Main Menu
```

## 2. Add function (enter '1')

(if input contain null, it output **null input try again**)

For **contact** we need to enter: **name, phone number, email, address**

```
Enter your choice (1-4): 1
Enter name: jack
Enter phone number: 12345
Enter email address: ppp@ll.com
Enter address: popop
```

For **event** we need to enter: **description, start time, alarm**

(note start time must follow format: YYYY-MM-DD HH:MM, and alarm indicate minutes)

```
Enter event description: noting
Enter event start time (YYYY-MM-DD HH:MM): 2023-11-11 11:11
Enter event alarm (minutes before start time): 16
```

For **event** we need to enter: **description, ddl**

```
Enter task description: task1
Enter event deadline (YYYY-MM-DD HH:MM): 2000-11-11 11:11
```

(note start time must follow format: YYYY-MM-DD HH:MM)

## 3. View function (enter '2')

(View of contacts)

```
Description: task1

DDL: 2000-11-11 11:11

-----

Description: popop

DDL: 2000-11-12 00:00

-----
```

(View of events)

```
Name: gcc-c
Phone: 198001
Email: popp@gg.com
Address: loxer
-----
Name: sadc
Phone: 3243
Email: 21431234@f.com
Address: adsfasdf
```

(View of tasks)

```
Description: dll1
DDL: 2000-11-12 00:00
-----
Description: pops
DDL: 2000-11-13 09:09
-----
```

#### 4. View delete (enter '3')

(if delete reference **not found**, it output **file not found**)

For **contact** we need to delete by **name**

```
Enter name of contact to delete: gcc-c
Contact deleted successfully.
```

For **event** we need to delete by **description**

```
Enter event description to delete: noting
Event deleted successfully.
```

For **task** we need to delete by **description**

```
Enter task description to delete: task1
Task deleted successfully.
```

#### 5. Back to main menu (enter '4')

back to main page

## 2.2 QuickNote

Enter '3' to enter note mode, This mode allows for line breaks. Typing 'END' followed by the Enter key signifies the end of input. At this point, four options will appear: enter 'w' to rewrite and overwrite the existing file, enter 'a' to append to the end of the file, enter 'c' to continue note-taking, and enter 'q' to exit without saving the file. If any other commands are entered, a message will prompt an incorrect input.

```
9. Exit
Enter your choice (1-4, 9): 3
would you like to name the note ? (y/n) > y
Enter filename start with 'QuickNotes' > QuickNotes_test
=====Take Quick Notes=====
=====Enter END to END=====
I
Like
Programming
END
Rewrite(w) | Append(a) | Continue(c) | Quit without save(q)
```

## 2.3 Search

For the search function, the system will directly retrieve matching files and instances in the database through the prompt words entered by the user and return them, allowing users to quickly obtain instance information. User need to first enter type of file they want to search (or 'q' to quit), then input further information. The output will be the list of the name of matching file. Invalid format of input (such as wrong format of operator with keywords, datetime format)

```
input type of pir you want to search quit(q):
```

For **Contacts** and **QuickNotes** we support search by **keywords** (also with '|', '&&', '!')

```
input type of pir you want to search quit(q): Contacts
input keywords (support !, ||, &&): vivi && Pep ! June
file find as follows:
['Contacts3.pim']
```

```
input type of pir you want to search quit(q): QuickNotes
input keywords (support !, ||, &&): saca && I
file not found
```

For **Contacts** and **QuickNotes** we support search by **keywords** (enter '1') and search by **datetime** (enter '2') (with '>', '<', '=') (also with '|', '&&', '!')

```
input type of pir you want to search quit(q): Tasks
1)search by keywords 2)search by DDL :
2
enter time constraint with <, >, = (YYYY-MM-DD HH:MM)(support || &&): > 1999-10-10 00:00 && < 2022-10-13 09:00
file find as follows:
['Tasks1.pim', 'Tasks2.pim', 'Tasks_2023-11-23.pim']
```

```
input type of pir you want to search quit(q): events
1)search by keywords 2)search by start time :
2
enter time constraint with <, >, = (YYYY-MM-DD HH:MM)(support || &&): > 2023-11-16 01:57
file not found
```

## 2.4 Load

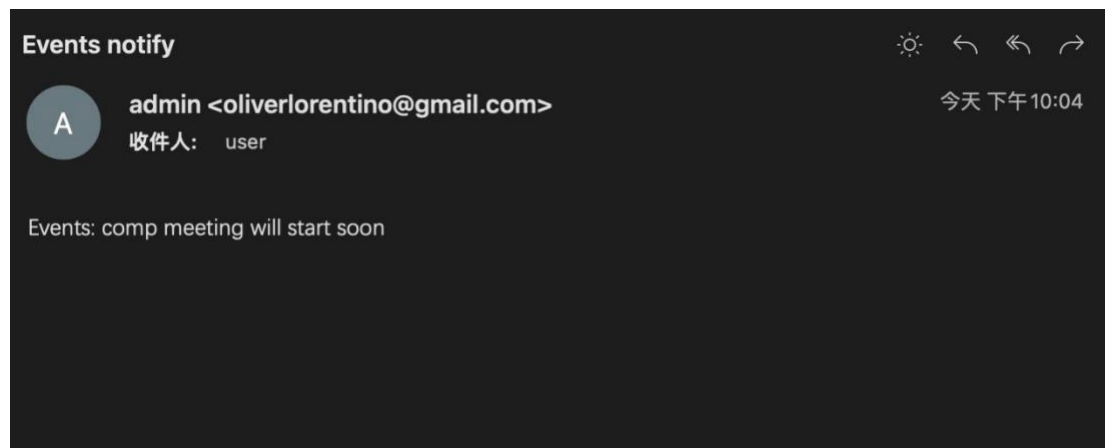
The load feature allows users to load a specified file into the system by entering the file name. If the file does not exist, it will return '**file not found**'. If the file exists, the system will access the file and provide relevant follow-up operations.

```
Enter your choice (1-4, 9): 5
enter the file name or 'q' to exit > Contacts1
format error : not end with pim
Contacts1 not exit in PIM_dbs
enter the file name or 'q' to exit > Contacts1.pim
=====load successfully=====
1. Add Contact
2. View Contacts
3. Delete Contact
4. Back to Main Menu
Enter your choice (1-4): |
```

## 2.5 Alarm

This is a hidden feature that does not require active usage. It starts a daemon process as soon as the program launches, reads events from the database, matches the current time with the event time and the preset alarm time, and sends notifications to the user via email.

```
Description: comp meeting
Start Time: 2023-11-23 22:05:00
Alarm: 0:01:00
-----
```



## 3. Precautions

When the user enters the file name, instance name, attribute name of the instance and searches for key words during use, the system presets the corresponding data type or string format. If the user does not follow the appropriate restrictions, the system will reject the user's input.

## 4. PIRs Input Restrictions

Mandatory field compliance is key; contacts require a name and at least one mode of contact, while events and tasks demand properly formatted dates and times.

## 5. Input Commands and Structure

The system operates on numerical input for menu navigation. Detailed prompts guide the entry of PIR-related information. Special commands (e.g., 'q' for quitting) are integral for navigation and exiting interfaces.

## 6. Invalid Input Handling

The system is designed to identify and alert users of incorrect inputs, prompting for rectification. Error messages are clear and directive, guiding towards correct data entry.

## 7. Output Interpretation

Upon successful operations, the system affirms actions with clear messages. Outputs, be they search results or viewing requests, are formatted for ease of comprehension.

## 8. Advanced Features

The search utility in the system is robust, supporting keyword-based, date-oriented, and logic-driven queries. Event alarms are automated, adhering to preset parameters for user convenience.