

PIM

User's Manual

Content

- 1. System startup and loading**
- 2. Note Model Operation Instructions**
- 3. Task Model Operation Instructions**
- 4. Event Model Operation Instructions**
- 5. Contact Model Operation Instructions**
- 6. Search instruction operation instructions**
- 7. Print Instruction Operating Instructions**
- 8. Invalid command outputs**

Overview

The PIM system implements comprehensive management functions for personal information records (PIR), supporting PIR types including notes, tasks, events, and contacts. Users shall load the xx.pim file to check their input PIRs.

1. System startup and loading

- Method 1: The default startup method of the system is to load the data.pim file under the subdirectory data in the current directory of the system (where pim.jar is located), with a relative path of (data/data.pim). If the data.pim file does not exist, it will be created as.java -jar pim.jar
- Method 2: Add the pim file directory parameter after the system startup command.

Starting this method will use the data file with this parameter to load. If the file does not exist, it will be created.
java -jar pim.jar data/data_new.pim

After the system starts, it will display the currently supported operating instructions:

```
=====Welcome to PIM system=====
note      list      List note
note      add       Add note
note      edit [id]  Edit note
note      delete [id] Delete note
task      list      List task
task      add       Add task
task      edit [id]  Edit task
task      delete [id] Delete task
event     list      List event
event     add       Add event
event     edit [id]  Edit event
event     delete [id] Delete event
contact   list      List contact
contact   add       Add contact
contact   edit [id]  Edit contact
contact   delete [id] Delete contact
search    [type|text|time] Search
print     [id|all]   Outputquer

-----
help      query list
quit      Exit PIM
Please input the operation: |
```

Among them, instructions `help` is the operation instructions. When user type `help`, the current user menu will be output again. Instructions `quit` is for exit the system.

Input Standards:

Here are some special input operation instructions for the system:

- i) Some fields support multi-line input, and the system will print out:

Exit the input operation when a line of input is `[: quit]`

```
-----
help      query list
quit      Exit PIM
Please input the operation:
note add
Please input content(When input equals to [:quit] Quit the system):
Hello!
:quit
save successfully!
```

This supports entering multiple lines of text. When a new line (after the Enter key) is entered, `'[:quit]'` will exit the current field input.

- ii) Some fields are stored as data types, and the system prompts you:

The format is YYYY-MM-DD, for example, on October 27, 2023

```
-----  
help          query list  
quit          Exit PIM  
Please input the operation:  
task add  
Please input deadline(format as:yyyy-MM-dd, eg.2023-10-27):  
2023-11-11  
Please input description(When input equals to [:quit] Quit the system):  
shopping  
:quit  
Save successfully!
```

Please enter the time information in this format.

2. Note Model Operation Instructions

- Add a note and enter the command:

```
note add  
Please input the operation:  
note add  
Please input content(When input equals to [:quit] Quit the system):  
Hello  
:quit  
save successfully!
```

- List notes, and enter commands:

```
note list  
Please input the operation:  
note list  
note list:  
Note{id='1177310751098929152', content='Hello'}
```

- Edit note, enter command:

```
note edit 1177310751098929152  
Please input the operation:  
note edit 1177310751098929152  
Please input content(When input equals to [:quit] Quit the system):  
Hello World  
:quit  
save successfully!
```

We can check whether the PIR was correctly edited. The parameter after the command is the id of the note.

```
Please input the operation:
note list
note list:
Note{id='1177310751098929152', content='Hello World'}
```

➤ Delete note, enter command:

```
note delete 1177310751098929152
```

3. Task Model Operation Instructions

➤ Add a task, and enter the command:

```
task add
Please input the operation:
task add
Please input deadline(format as:yyyy-MM-dd, eg.2023-10-27):
2023-11-25
Please input description(When input equals to [:quit] Quit the system):
Software due
:quit
Save successfully!
```

➤ List tasks, and enter commands:

```
task list
Please input the operation:
task list
task list:
Task{id='1177310810259587072', deadline='2023-11-25', description='Software due'}
```

➤ Edit task, and enter commands:

```
task edit 1177310810259587072
Please input the operation:
task edit 1177310810259587072
Please input deadline(format as:yyyy-MM-dd, eg.2023-10-27):
2023-11-27
Please input description(When input equals to [:quit] Quit the system):
Software Pre
:quit
Save successfully!
```

We can check whether the PIR was correctly edited. The parameter after the command is the id of the task.

```
Please input the operation:
task list
task list:
Task{id='1177310810259587072', deadline='2023-11-27', description='Software Pre'}
```

➤ Delete task, enter command:

```
task delete 1177310810259587072
```

4. Event Model Operation Instruction

➤ Add an event and enter the command:

```
event add
Please input the operation:
event add
Please input startTime(format as:yyyy-MM-dd, eg.2023-10-27):
2023-11-22
Please input alarm(format as:yyyy-MM-dd, eg.2023-10-27):
2023-12-05
Please input description(When input equals to [:quit] Quit the system):
Math
:quit
Save successfully!
```

➤ List events, enter commands :

```
event list
Please input the operation:
event list
event list:
Event{id='1177311668032503808', startTime='2023-11-22', alarm='2023-12-05', description='Math'}
```

➤ Edit event, enter command:

```
event edit 1177311668032503808
Please input the operation:
event edit 1177311668032503808
Please input startTime(format as:yyyy-MM-dd, eg.2023-10-27):
2023-11-13
Please input alarm(format as:yyyy-MM-dd, eg.2023-10-27):
2023-12-06
Please input description(When input equals to [:quit] Quit the system):
Math A2
:quit
Save successfully!
```

➤ We can check whether the PIR was correctly edited. The parameter after the command is the id of the task.

```
Please input the operation:
event list
event list:
Event{id='1177311668032503808', startTime='2023-11-13', alarm='2023-12-06', description='Math A2'}
```

➤ Delete event, enter command:

```
event delete 1177311668032503808
```

5. Contact Model Operation Instructions

- Add a new contact and enter the command:

```
contact add
Please input the operation:
contact add
Please input name:
Bob
Please input address:
Homantin
Please input mobile:
1234567
Save Successfully!
```

- List contacts, and enter commands :

```
contact list
Please input the operation:
contact list
contact list:
Contact{id='1177313367900028928', name='Bob', address='Homantin', mobile='1234567'}
```

- Edit contact and enter commands:

```
contact edit 1177313367900028928
Please input the operation:
contact edit 1177313367900028928
Please input name:
Andy
Please input address:
Hung Hom
Please input mobile:
7654321
Save Successfully!
```

- We can check whether the PIR was correctly edited. The parameter after the command is the id of the task.

```
Please input the operation:
contact list
contact list:
Contact{id='1177313367900028928', name='Andy', address='Hung Hom', mobile='7654321'}
```

- Delete contact, enter command:

```
contact delete 1177313367900028928
```

6. Search instruction operation instructions

- Search instruction search:

```
search [type|text|time]
```

Instructions:

Query command format “search + blank space + optional” conditions

The optional conditions are “type | text | time”, and the combined conditions must be in this order. Type represents the model type, text searches for text fields in the model, time searches for date data in the model, and compares the conditions using greater than, less than, or equal methods.

The value of the condition can be added before “!”, Represents the negation operation.

Multiple conditions can be used “&&” or “||”.

When there is no condition and only search is executed, all PIR records in the system will be queried.

Example explanation:

➤ Query all data: search

```
-----
help          query list
quit          Exit PIM
Please input the operation:
search
Note{id='1177920473451724800', content='Hello World'}
Note{id='1178061093352767488', content='Hello!'}
Task{id='1177321876091830272', deadline='2023-11-27', description='software presentation'}
Task{id='1177322111916572672', deadline='2023-12-31', description='Hello 2024'}
Task{id='1178061216120045568', deadline='2023-11-11', description='2023-12-12shopping'}
Event{id='1177688321799749632', startTime='2023-11-23', alarm='2023-12-6', description='Hello Math A2'}
Event{id='1178061407472582656', startTime='2022-1-1', alarm='2023-1-1', description='annual'}
Contact{id='1177688176509059072', name='Bob', address='Homantin', mobile='1234567'}
Contact{id='1178061578847649792', name='Andy', address='Humhung', mobile='1122334'}
```

➤ Query data of type note:

```
-----
help          query list
quit          Exit PIM
Please input the operation:
search type=note
Note{id='1177920473451724800', content='Hello World'}
Note{id='1178061093352767488', content='Hello!'}
```

➤ Query data with a type other than note:

```
search type!=note
Please input the operation:
search type!=note
Task{id='1177310810259587072', deadline='2023-11-27', description='Software Pre'}
Task{id='1177311163256406016', deadline='2023-11-23', description='Math Start'}
Task{id='1177311462813597696', deadline='2023-12-31', description='Hello 2024'}
Event{id='117731168032503808', startTime='2023-11-13', alarm='2023-12-06', description='Math A2'}
```

- The query type is not note, and the text contains “Hello”:

```
search type=!note && text=Hello
```

```
Please input the operation:
```

```
search type=!note && text=Hello
```

```
Task{id='1177311462813597696', deadline='2023-12-31', description='Hello 2024'}
```

- The query type is not note, or the text does not contain “Math”:

```
search type=!note || text=!Math
```

```
Please input the operation:
```

```
search type=!note || text=!Math
```

```
Note{id='1177310751098929152', content='Hello World'}
```

```
Task{id='1177310810259587072', deadline='2023-11-27', description='Software Pre'}
```

```
Task{id='1177311462813597696', deadline='2023-12-31', description='Hello 2024'}
```

```
Contact{id='1177313367900028928', name='Andy', address='Hung Hom', mobile='7654321'}
```

- The query type is task, and the time>2023-11-15:

```
Search type=task && time>2023-11-15
```

```
-----  
help          query list
```

```
quit          Exit PIM
```

```
Please input the operation:
```

```
search type=task && time>2023-11-15
```

```
Task{id='1177321876091830272', deadline='2023-11-27', description='software presentation'}
```

```
Task{id='1177322111916572672', deadline='2023-12-31', description='Hello 2024'}
```

7. Print instruction operation instructions

- Output instruction print:

Instructions:

Query command format: “print + blank space + id” or simply “print”

All indicates that the query outputs all (the same as the search instruction without parameters)

```
help          query list
```

```
quit          Exit PIM
```

```
Please input the operation:
```

```
print all
```

```
Note{id='1177920473451724800', content='Hello World'}
```

```
Note{id='1178061093352767488', content='Hello!'}
```

```
Task{id='1177321876091830272', deadline='2023-11-27', description='software presentation'}
```

```
Task{id='1177322111916572672', deadline='2023-12-31', description='Hello 2024'}
```

```
Task{id='1178061216120045568', deadline='2023-11-11', description='2023-12-12shopping'}
```

```
Event{id='1177688321799749632', startTime='2023-11-23', alarm='2023-12-6', description='Hello Math A2'}
```

```
Event{id='1178061407472582656', startTime='2022-1-1', alarm='2023-1-1', description='annual'}
```

```
Contact{id='1177688176509059072', name='Bob', address='Homantin', mobile='1234567'}
```

```
Contact{id='1178061578847649792', name='Andy', address='Humhung', mobile='1122334'}
```

Or print a specific PIR with its id.


```

-----
help          query list
quit          Exit PIM
Please input the operation:
print 1178061578847649792
Contact{id='1178061578847649792', name='Andy', address='Humhung', mobile='1122334'}

```

8. Invalid command outputs

When editing a PIR without id parameter being specified, the system prompts

When executing the edit command, please include a unique identifier id after the command

Eg. For note edit.

```

-----
help          query list
quit          Exit PIM
Please input the operation:
note edit
When executing the Edit command, please attach a unique id

```

When the record corresponding to the specified ID does not exist, the system prompts:

The specified note does not exist!

Eg. For note edit.

```

-----
help          query list
quit          Exit PIM
Please input the operation:
note edit 1212121
The note does not exist!

```

When the search condition is wrong, the system will prompt:

The type query condition entered does not meet the requirements, please re-enter it!

Eg. Search by type but without specific type.

```

-----
help          query list
quit          Exit PIM
Please input the operation:
search type=
The type query condition entered does not meet the requirements, please re-enter it!

```