

# PIM - User Manual

---

**Project Name:** Personal Information Manager

**Group Number:** 13

**Group Member:**

- JIANG Guanlin (21093962D)
  - MENG Guanlin (20099185D)
  - HU Yuhang (21106395D)
  - YE Feng (21098249D)
- 

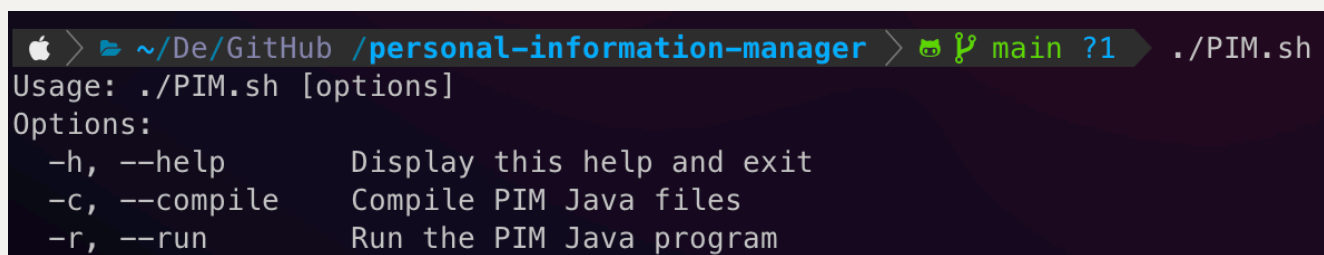
## 1 Introduction

Personal information manager (PIM) is a system which allows users to manage their personal information conveniently. First, users should sign up and log in an account to manage their personal information. All the personal information of a user can be recorded as four types of personal information records (PIRs) which are contacts, notes, tasks and events. In addition to creating new PIRs, the system can also support other four functions about PIRs, including printing out detailed information about PIRs, modifying PIRs, deleting PIRs, searching for PIRs based on keywords, time, multiple keywords and/or time with logical connectors. For all functions about PIRs, there is a restriction that the option input by the user can only operate one PIR but cannot operate several more PIRs simultaneously. For removing PIRs, there is a restriction that there is no option for users to delete all PIRs.

## 2 PIM Starting

### 2.1 PIM Program Help Command

```
./PIM.sh -h  
## OR  
./PIM.sh --help  
## OR  
./PIM.sh
```



A terminal window screenshot showing the execution of the `./PIM.sh` command. The prompt is `main ?1`. The output displays the usage and options for the script.

```
Apple > ~/De/GitHub /personal-information-manager > main ?1 ./PIM.sh  
Usage: ./PIM.sh [options]  
Options:  
-h, --help          Display this help and exit  
-c, --compile        Compile PIM Java files  
-r, --run            Run the PIM Java program
```

### 2.2 PIM Program Compile Command

```
./PIM.sh -c  
## OR  
./PIM.sh --compile  
## OR  
./PIM.sh
```

### 2.3 PIM Program Running Command

```
./PIM.sh -r  
## OR  
./PIM.sh --run
```

## 3 PIM Program Usage

### 3.1 Initialization Interface

```
-----  
Welcome to the Personal Information Manager (PIM)!  
COMP3211 - Software Engineering  
Project Group 13  
  
- [1] Login  
- [2] Signup  
- [-1] Exit System  
  
>>> Please select the above options x in [x]:
```

### 3.2 User Signup

```
-----  
Signup  
  
Username: david  
Password: 1111  
-----
```

- Input: Username, Password
- Output: Let the user to input the informations (Guide)

### 3.3 User Login

```
-----  
Login  
  
Username: david  
Password: 1111  
Login Success!  
-----
```

- Input: Username, Password
- Output: Let the user to input the informations (Guide)

### 3.4 Main Menu Interface

```
-----  
Welcome david  
-----  
  
- [1] Notes  
- [2] Contacts  
- [3] To-do Lists  
- [4] Events  
- [5] Load .PIM File  
- [6] Export .PIM File  
- [-1] Exit System  
  
>>> Please select the above options x in [x]: |
```

- Input: Select Number
- Output: Main Menu View

## 3.5 Note View Interface

```
-----  
Notes - david  
  
- [1] New Notes  
- [2] Read Note  
- [3] Search Note  
- [4] Modify Note  
- [5] Remove Note  
- [-1] Back to last page
```

```
>>> Please select the above options x in [x]: |
```

- Input: Select Number
- Output: Note Menu View

### 3.5.1 Create Note Interface

```
>>> Please select the above options x in [x]: 1
```

```
Note Title: Note 1
```

```
Note Content:
```

```
-----  
Here is Note 1 and Hello World!  
-----
```

```
- [1] Save Note  
- [2] Discard Note
```

```
>>> Please select the above options x in [x]: 1
```

- Input: Note Title, Note Content, Select Number
- Output: Let the user to input the informations (Guide)

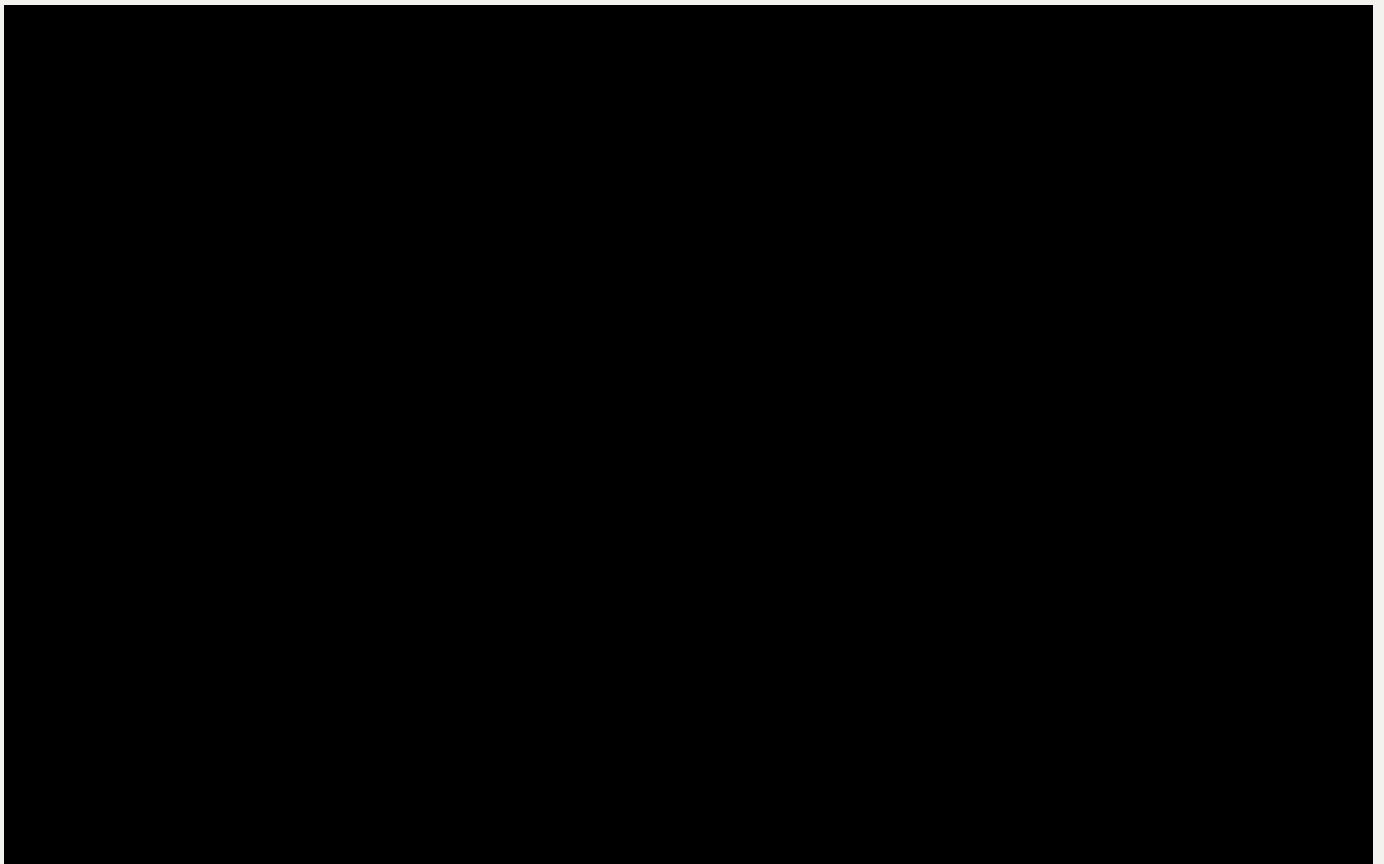
### **3.5.2 Read Notes Interface**

- Input: Note ID, Select Number
- Output: User selected Note ID will be printout all the notes, and if select all, all the notes will be printout

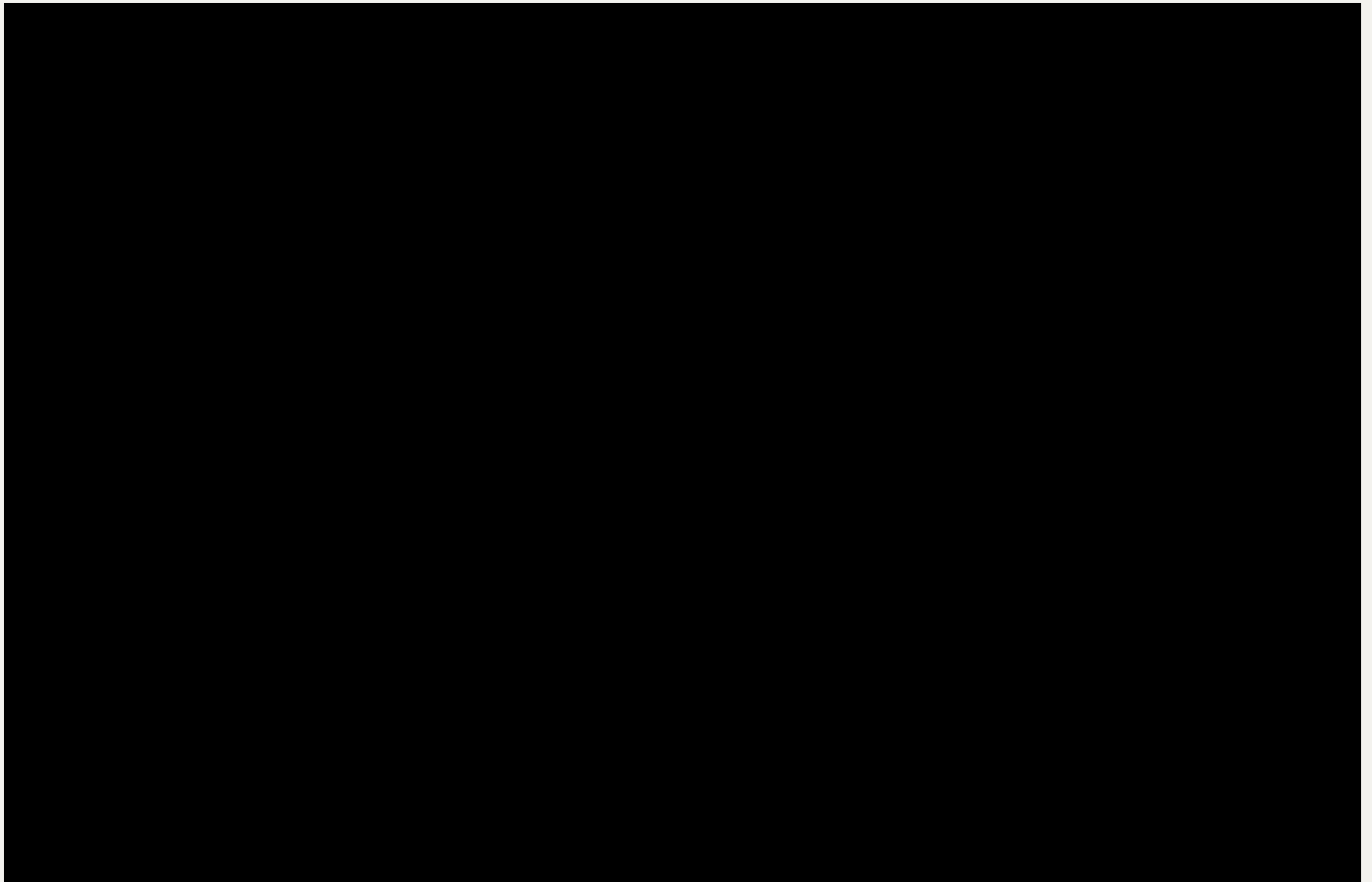
### **3.5.3 Search Note Interface**

#### **3.5.3.1 Search by Keyword**

#### **3.5.3.2 Search by Time**



#### **3.5.3.3 Search by Logic Connector**



- Input: Searching Keyword(s) or Time or Keywords with logical Connector
- Output: The results of the searching

### **3.5.4 Modify Note Interface**

- Input: Note ID, Note Title, Note Content
- Output: The If the update successful, will printout 'Update Successfully!'

### **3.5.5 Remove Note Interface**

- Input: Note ID, Select Number
- Output: If the remove successful, the 'Remove Successfully!'

## 3.6 Contact View Interface

- Input: Select Number
- Output: Contact Menu View

### 3.6.1 Create Contact Interface

- Input: First Name, Last Name, Phone Number, Address, Select Number
- Output: Let the user to input the informations (Guide)

### 3.6.2 All Contacts Interface

- Input: Contact ID, Select Number
- Output: User selected Contact ID will be printout all the contacts, and if select all, all the contacts will be printout

### 3.6.3 Search Note Interface

#### 3.6.3.1 Search by Keyword

#### 3.6.3.2 Search by Logic Connector

- Input: Searching Keyword(s) or Date or Keywords with logical Connector
- Output: The results of the searching



### **3.6.4 Modify Contact Interface**

- Input: Contact ID, First Name, Last Name, Phone Number, Address, Select Number
- Output: The If the update successful, will printout 'Update Successfully!'

### **3.6.5 Remove Contact Interface**

- Input: Contact ID, Select Number
- Output: If the remove successful, the 'Remove Successfully!'

## **3.7 To-do View Interface**

- Input: Select Number
- To-do Task Menu View

### **3.7.1 Create Task Interface**

- Input: Task Name, Task Deadline, Task Description, Select Number
- Output: Let the user to input the informations (Guide)

### **3.7.2 All Tasks Interface**

- Input: Task ID, Select Number
- Output: User selected Task ID will be printout all the tasks, and if select all, all the tasks will be printout

### **3.7.3 Search Task Interface**

#### **3.7.3.1 Search by Keyword**

#### **3.7.3.2 Search by Time**

#### **3.7.3.3 Search by Logic Connector**

- Input: Searching Keyword(s) or Time or Keywords with logical Connector
- Output: The results of the searching

### **3.7.4 Modify Task Interface**

- Input: Task ID, First Name, Last Name, Phone Number, Address, Select Number
- Output: The If the update successful, will printout 'Update Successfully!'

### **3.7.5 Remove Task Interface**

- Input: Task ID, Select Number
- Output: If the remove successful, the 'Remove Successfully!'

## **3.8 Event View Interface**

- Input: Select Number
- Output: Events Menu View

### **3.8.1 Create Event Interface**

- Input: Event Name, Event Start Time, Event Alarm, Event Description, Select Number
- Output: Let the user to input the informations (Guide)

### **3.8.2 All Events Interface**

- Input: Event ID, Select Number
- Output: User selected Event ID will be printout all the events, and if select all, all the events will be printout

### **3.8.3 Search Event Interface**

#### **3.8.3.1 Search by Keyword**

#### **3.8.3.2 Search by Time**

#### **3.8.3.3 Search by Logic Connector**

- Input: Searching Keyword(s) or Time or Keywords with logical Connector
- Output: The results of the searching

### **3.8.4 Modify Event Interface**

- Input: Event ID, Event Name, Event Start Time, Event Alarm, Event Description, Select Number
- Output: The If the update successful, will printout 'Update Successfully!'

### **3.8.5 Remove Event Interface**

- Input: Event ID, Select Number
- Output: If the update successful, the 'Update Successfully!'

### **3.9 Import PIM file Interface**

- Input: Select Number
- Output: If successful, will printout 'import Successfully!'

### **3.10 Export PIM file Interface**

#### **Export File Sample:**

- Input: Select Number
- Output: If successful, will printout 'Export Successfully!'

## **4 Error Handling**

### **4.1 Selection Input Error Handle**

- Give the error message, and try again.

## **4.2 Login Error Handle**

- Give the error message and the program will close, so run the program again.

## **4.3 Input Date Error Handle**

- Give the error message, and rewrite again.

## **4.4 Searching no result Handle**

- When searching and no result, will be return the empty.