Software Requirements Specification (SRS) for Personal Information Manager (PIM)

COMP3211 Software Engineering Project

Group 26

Han Wenyu	21097519D
Li Yangxiaoxuan	20102514D
Tam Chung Man	21073064D
Zhou Siyu	21094655D

Content

Preface

Expected Relationship: Software User, Software Manager, Investors, Software Developer

Version: 1.0 beta

Updates: N/A

Change of each version: N/A

Introduction

Here is the structure of requirements specification for a command-line-based personal information manager (PIM). This document aims to provide a comprehensive guide for how to build well-organized requirements specification for the personal information manager (PIM), which will be helpful for software developers and users to properly understand and articulate the requirements.

The command-line-based personal information manager (PIM) allowed users to create different types of personal information records (PIRs), including task, note, event, and contacts. Each information in a single location, and can be modified, searched, printed, deleted, and loaded in the following commands input.

In today's fast-paced digital world, effectively handling personal information has become more essential, a user-friendly command-line-based personal information manager enabling users to simply and flexible access and organize their schedules, to maximize their productivity and better use their time.

The PIM system is designed to grow with the users' changing needs, it can integrate with other systems to further explore the flexibility. In the future, it enables sharing tasks to effectively cooperate among team members, by integrating with other collaboration tools. Both personal and team's schedule can be uploaded to the calendar for future daily planning, the information of alarms and contacts also can connect with email for sending notices and communicating with others, respectively.

Glossary

In this part, There will be few technical terminology in this document and corresponding abbreviations for these terms, which helps reader better understanding the document.

Abbreviation	Meaning
CLI	Command Line Interface
MVC	Model-View-Controller Architecture
PIR	Personal Information Record
PIM	Personal Information Manager

Table 1 Abbreviation Table for this document

User Requirements Definition

User Functional Requirements

1. Users shall create PIRs and save in a single location based on type of PIRs.
2. Users shall create plain texts as PIRs to take quick notes, under the hints from the system about input specific commands.
3. Users shall create tasks as PIRs to manage to-dos, including description and deadlines, under the hints from the system about input specific commands.
4. Users shall create events as PIRs to manage schedule, including description, starting time and alarm, under the hints from the system about input specific commands.
5. Users shall create contacts as PIRs to manage contacts, including names, addresses, and mobile numbers, under the hints from the system about input specific commands.
6. The created PIRs shall be modified by user inputting a command, under the hints from the system about input specific commands.
7. Users shall search specific PIRs based on text criteria, checking if the text fields contain a specified string.
8. Users shall search specific PIRs based on time criteria, specifying whether the time is before (<), after (>), or equal to (=) another given point in time.
9. Users shall search specific PIRs based on logical criteria: AND (&&), OR (||), and negation (!), and do complex searches combining different criteria using connectors.
10. Users shall print out detailed information of the specific PIRs to review, under the hints from the system about input specific commands.
11. Users shall delete the specific PIRs, under the hints from the system about input specific commands. Confirmation prompt should be displayed to the user for double check.
12. Users shall store PIRs by inputting a file name with the extension name “.pim”, under the hints from the system about input specific commands. Confirmation and success save prompt should be displayed to the user for double check.
13. Users shall load PIRs from files to edit detailed information, under the hints from the system about input specific commands. Last modified time should display as reference for the user.

User Non-functional Requirements

1. The personal information manager runs as a command-line system.
2. The process of creating and managing PIRs shall be user-friendly, with clear hints to guide the users.
3. The PIM shall safely and accurately save all created PIRs, without any data leakage or loss, saving time shall not exceed 10 seconds.
4. The printed information of PIRs should be well-formatted and organized for easy readability, loading time shall not exceed 10 seconds.

System architecture

In this chapter, we will present a high-level overview of the anticipated system architecture for our application. The system follows the Model-View-Controller (MVC) design pattern, with the Model responsible for data and business logic, the View handling user interface and presentation, and the Controller managing the flow of data and operations between the Model and the View. The reusability of architectural components enhances modularity and maintainability, making the system more flexible and scalable. In the following chapters, we will delve deeper into each component and their interactions to provide a comprehensive understanding of the system architecture.

High-level overview of system architecture

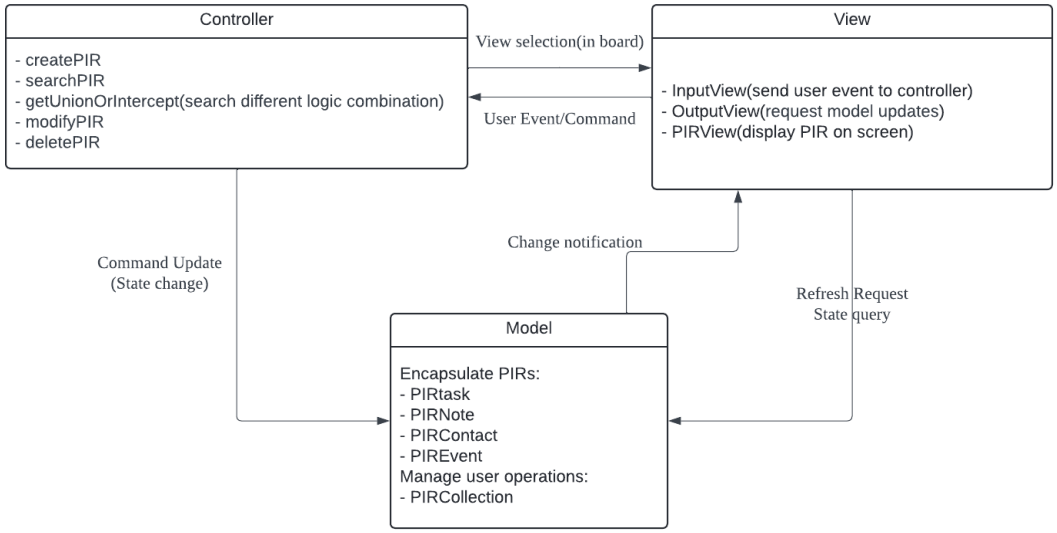

Figure 1 Model-View-Controller Diagram for PIM

The overall architecture is divided into three parts including model, view, and controller(See the Figure above). For the Model part, it contains encapsulated PIRs (PIRTask, PIRNote, PIRContact, PIREvent) and user operations of all types of PIR (PIRCollection). For the View part, it mainly contains InputView(receiving users' operation), OutputView(request model update, hints), PIRView(display PIR on screen) and other visualized contents with interface for action in PIR creation, search, modification, deletion. For the Controller part, it mainly receives users' operation with interface in View parts, and turns command into logic, then does the operation to PIR using the model.

Distribution of functions across system modules

Model

The Model component forms the backbone of the system and is responsible for managing the data and business logic. It encapsulates the core functionalities and represents the state of the application.

- **PIRNote**

The PIRNote component represents a Personal Information Record (PIR) in the form of a note. It provides methods for initializing the note, setting the note content, getting note information, and converting the note to string and PIR formats.

- **PIRTask**

The PIRTask component represents a PIR in the form of a task. It allows for initializing the task, setting the task description and deadline, retrieving task information, and converting the task to string and PIR formats.

- **PIRContact**

The PIRContact component represents a PIR in the form of a contact. It enables the initialization of contact information, setting the contact's name, address, and mobile number, retrieving contact information, and converting the contact to string and PIR formats.

- **PIREvent**

The PIREvent component represents a PIR in the form of an event. It facilitates the initialization of event details, setting the event description, start time, and alarm, retrieving event information, and converting the event to string and PIR formats.

- **PIRCollection**

PIRCollection components represent different types of function managing and manipulating different types of PIR records within the system. It facilitates in method for searching (matching time and text), filtering(index of lines for searched PIR records), replace function for modification or updating of PIR records, and deleting PIR records, insert function for creation of PIR records.

View

The **View** component is responsible for presenting information to the user and handling user interactions. It provides an user interface to interact with the system and receives input from them.

- ***InputView***

The *InputView* module receives user operations and input. It captures the commands and data entered by the user and passes them to the *Controller* for further processing.

- ***OutputView***

The *OutputView* module is responsible for displaying output to the user. It handles requests from the Model for updating the view and provides hints or guidance to the user based on the system state.

- ***PIRView***

The *PIRView* module displays the Personal Information Records (PIRs) on the screen. It presents the PIR information in a visually appealing and user-friendly manner, allowing users to view and interact with their records.

Controller

- ***PIRController***

The *PIRController* component acts as an intermediary between the *Model* and the *View*, and it also update database in model after processing command input from user. This method functionally provides a home page directed into different operations: creation of PIR, search of PIR, deletion of PIR, display of PIR.

Reusability in Architecture

Reusability in Model

PIRNote, PIRTask, PIRContact, PIREvent, PIRCollection modules in Model block which encapsulates the data, and operations related to different kinds of PIR records. The operation and the data can be reused whenever there is a need to handle *Note/Task/Contact/Event* functionality. Whether it is creating, modifying, or deleting *PIR*, operation functions in these modules provides a reusable solution that can be easily integrated into different parts of the system.

Reusability in View

InputView, OutputView, PIRView modules provides different user interfaces for user interaction. For example, *InputView* module can be reused across multiple modules to collect user input following a standard routing. Also, *OutputView* module can be reused when displaying information and hints following a standard routing. At the same time, *PIRView* module can be reused when displaying PIR records on the screen. Whenever it is using functions relating to interaction with users, functions in these modules provides a reusable solution that can be easily integrated into different parts of the PIM system.

System Requirements Specification

System Functional Requirements

No.	1
Title	Select PIRs type
Requirements	The PIM shall provide hints for users about how to input commands to choose for creating different types of PIRs. The PIM shall save each PIRs in a single location and have specific fields based on PIRs' type.
Rationale	Select the type of PIRs displayed on the menu, more clear and convenient for users entering corresponding classes to input detailed information.
Reference	UR1

No.	2
Title	Plain texts creation
Requirements	The PIM shall provide hints for users about how to input commands to create new plain texts as PIRs, inputting descriptions to take quick notes.
Rationale	To satisfy user's need to take quick notes.
Reference	UR2

No.	3
Title	Tasks creation
Requirements	The PIM shall provide hints for users about how to input commands to create new tasks as PIRs, inputting description and specific time and dates as deadline to manage to-dos. The PIM should define the structure of inputting detailed information about a task.
Rationale	To satisfy user's need to manage to-dos with specific structure, help schedule management become more efficient.
Reference	UR3

No.	4
Title	Events creation
Requirements	The PIM shall provide hints for users about how to input commands to create new events as PIRs, inputting description and specific time and dates as starting time and alarms to manage schedules. The PIM should provide various alarms methods to users, which is suitable for different types of alarms. The PIM shall save the created events.
Rationale	To satisfy user's need to manage schedules more efficiently. Ensuring both date and time components for starting time is specific enough, both times and method components for alarm, to meet the different users' needs for alarm.
Reference	UR4

No.	5
Title	Contacts creation
Requirements	The PIM shall provide hints for users about how to input commands to create new contacts as PIRs, inputting name, address, and mobile numbers. The PIM shall check the format is valid for different types of contact's information. The PIM shall save the created contacts.
Rationale	To satisfy user's need to manage contact information. Checking whether inputting contact information matches the requirements for various formats, including numeric and textual.
Reference	UR5

No.	6
Title	Modify PIRs
Requirements	The PIM shall provide hints for users about how to input commands to modify different data from various types of existing PIRs.
Rationale	The PIRs may need to be modified due to different reasons, so users need systematic guidelines on how to modify various types of specific information in PIM.
Reference	UR6

No.	7
Title	Select PIRs
Requirements	The PIM shall search specific PIRs based on text criteria, including corresponding fields of PIRs, such as description and name, checking if the text fields contain a specified string. The PIM shall search specific PIRs based on time criteria, including corresponding fields of PIRs, such as deadline and starting time, specifying whether the time is before (<), after (>), or equal to (=) another given point in time. The PIM shall do search specific PIRs based on logical criteria: AND (&&), OR (||), and negation (!), and do complex searches combining different criteria using connectors. The PIM should allow creating complex search queries by combining different criteria using logical operators.
Rationale	Help users complete more flexible and comprehensive searches from text, time and logic criteria, including searches with complex conditions.
Reference	UR7, UR8, UR9

No.	8
Title	Print PIRs
Requirements	The PIM shall provide hints for users about how to input commands to print out the specific PIRs or all PIRs, including all relevant fields and attributes.
Rationale	Help users to clearly review detailed information about existing PIRs.
Reference	UR10

No.	9
Title	Delete PIRs
Requirements	The PIM shall provide hints for users about how to input commands to delete the specific PIRs. Before deleting the PIRs, the PIM should display a confirmation prompt to the user, including relevant details of the PIRs, to ensure the user is deleting the intended PIR. The PIM should display appropriate error messages if any issues are encountered while deleting the PIR, such as invalid identifier or deletion failures.
Rationale	After confirming that it is not deleted by mistake, to satisfy user's need to delete PIRs which are no longer required.
Reference	UR11

No.	10
Title	Store file
Requirements	The PIM shall provide hints for users about how to input commands to enter the file name with the extension name “.pim”. The PIM should allow the user to confirm before store, and remain the user when the file is stored successfully.
Rationale	To satisfy user’s need to store a file with a fixed suffix name and confirmation.
Reference	UR12

No.	11
Title	Load file
Requirements	The PIM shall provide hints for users about how to input commands to load a file name with the extension name “.pim”, and continue editing the latest version. The PIM should display the last-modified time. The PIM should make a warning when the user enters a wrong file name.
Rationale	The files may need to be modified due to different reasons, to satisfy user’s need to continue editing. Modified time and error message display are helpful hints for finding and modifying the loading file.
Reference	UR13

System Non-functional Requirements

No.	1
Title	Usability
Requirements	The PIM shall have user-friendly and logical instructions for inputting commands, which is clear and concise for the user to navigate and understand. The PIM shall provide well-formatted and organized output, which is easy for the user to read and understand. The PIM shall be case-insensitive to provide flexibility in handling different operations of PIRs.

No.	2
Title	Performance
Requirements	The PIM shall respond quickly to the user's command within 5 seconds, reducing latency and guaranteeing a responsive and smooth user experience. The PIM should quickly retrieve and display detailed information from the PIRs storage within 5 seconds, reducing noticeable delays.

No.	3
Title	Reliability
Requirements	The PIM shall safely and accurately save all created PIRs, without any data leakage or loss. The PIM shall provide a clear error message within 3 seconds, to help users identify the exact cause of the issues and provide solutions in time.

No.	4
Title	Maintainability
Requirements	The PIM should be developed using well-structured code, easy to understand for different developers to add new functions in the future.