

PERSONAL INFORMATION MANAGER

Software Requirement Specification

GROUP 69

Members:

1. NIYITEGEKA Berwa Aime Noel, 21104645D
2. LEE Jer Tao, 21098286D
3. LI Tong, 21101988D
4. Liu Yuzhou, 21100602D

Preface

Audience: Application users, Software Managers, Software Engineers

Version: 1.0

Updates: N/A

Change Rationale: N/A

INTRODUCTION

Purpose

This is the software requirement document (SRS) for the application Personal Information Manager (PIM). PIM is an application to help users manage and organize their personal information in one place. PIM aims to simplify personal information management and enhance organization for users.

Scope

The scope of development of Personal Information Manager (PIM) is development, documentation and testing of a command line application with the following features:

- **Creation:** Users can create various types of personal information records (PIRs) such as notes, tasks, events, and contacts.
- **Modification:** Users can update and modify existing PIRs to keep them up to date.
- **Search:** Users can search for PIRs based on specific criteria, including text in notes, descriptions, names, addresses, or mobile numbers, as well as time-based conditions using logical connectors and negation.
- **Printing:** Users can print detailed information about specific PIRs or all PIRs.
- **Deletion:** Users can delete specified PIRs from the application.
- **File Management:** PIRs can be stored in ".pim" files for future access and loaded from these files.
- **Command line interface:** users can interact with the application in the command line interface provided by their respective operating systems.

Glossary

This section defines the technical terminology in this document. As below, on the left ‘Term’ represents the abbreviation of the term, while ‘Meaning’ represents the acronym of the respective abbreviation.

Term	Meaning
PIM	Personal Information Manager
PIR	Personal Information Record
UID	Unique Identifier
RI	Requirement Index
RRI	Related Requirement Index
CV	Contact viewer
TV	Task viewer
EV	Event viewer
NV	Note viewer

User requirements

Index	Description
R1	Users shall be able to create different types of personal information record (PIR) in the personal information manager (PIM).
R2	Users shall be able to manage all information within a single location.
R3	Users shall be able to create new plain texts.
R4	Users shall be able to create new tasks with the corresponding descriptions, and deadlines as PIRs so that they are able to manage my schedule.
R5	Users shall be able to create new contacts with names, addresses, and mobile numbers.
R6	Users shall be able to modify data in existing PIRs.
R7	Users shall be able to search for PIRs based on different criteria.
R8	Users shall be able to print out detailed information about a specific PIR or all PIRs.
R9	Users shall be able to delete a specific PIR.
R10	Users shall be able to store the PIRs in a file with the extension name ".pim".
R11	Users shall be able to load the PIRs from a file with the extension name ".pim".

Non-functional System

Index	Description
R12	The system shall be accessed based on the command-line terminal.
R13	Every PIR has its own UID and the UID shall not be modified.
R14	The system shall check whether prior records exist before performing certain operations, such as searching, deleting, and updating.
R15	The system shall display the latest information after users perform certain operations, such as creating, deleting, and updating.
R16	The user shall be able to add, update, or delete alarm of events.

System architecture

Model-View-Controller (MVC) Architecture is implemented in this system. It is based on the idea that an application consists of three main logical components: the model, the view, and the controller.

Reason of selecting MVC Architecture:

1. **Easy to Division:** By dividing our system into Model, View, and Controller three components, it will benefit us as this approach will make the overall structure of the system clear and easy to understand. By separating the interface and I/O, core computation procedures, and status management or control, error rate can be decreased and the development process could be smoother.
2. **Development Friendly:** Having a well-defined architecture, such as the MVC (Model-View-Controller) pattern will greatly facilitate collaborative development. It eliminates the need for lengthy discussions and debates about the system's structure and how to divide the work. Instead, we can focus on establishing clear communication among the different modules, using tools like UML diagrams, and ensuring the system runs without any bugs.
3. **Development Experience:** We have experience implementing the MVC architecture during our previous OOP Group Project, and this experience will greatly benefit us by allowing us to work and prioritize tasks more effectively. Furthermore, since we plan to continue using MVC in our future studies and work, this project provides an excellent opportunity to accumulate experience and improve our skills.

The functions:

1.View

View is about the Interface and I/O:

Different view components are used, such as Contact Viewer (CV), Task Viewer (TV), Note Viewer (NV), Event Viewer (EV), to display different types of PIRs to users according to the most appropriate view component.

2. Controller

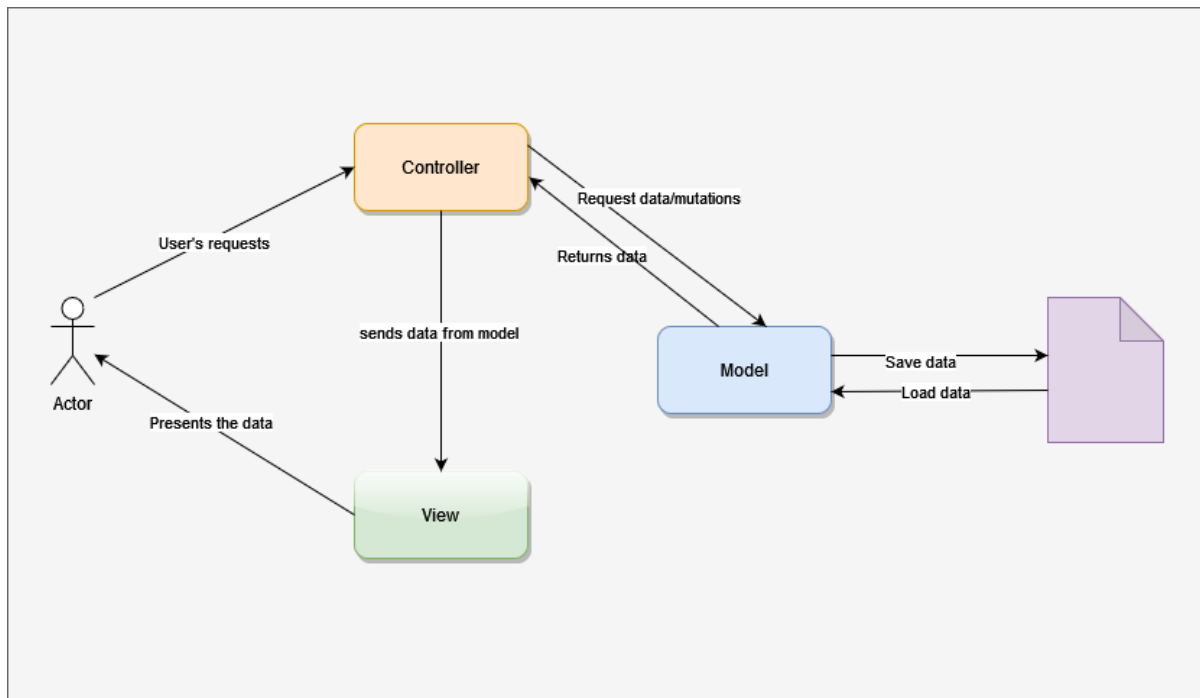
The manager of the whole system. The controller may control which mode the user enters. For example, when the user chooses to enter contact mode, the user will edit information in the file, which is called contacts.pim, and utilize the functions associated with contacts.

3. Model

The back end, computation core, and the recorder of the system.

Note, Event, Contact, Task: the entity for different types of PIRs. To record different kind of descriptions.

NoteManager, ContactManager, TaskManager, EventManager: The manager entity includes methods to manage different types of PIRs, such as create, update, delete, and search.



System requirements specification

This section describes functional and non-functional requirements in more detail and further details will be added to the non-functional requirements.

System Functional Requirements

For below, if not stated explicitly, the methods will be able to share among different types of PIRs.

ID	SR1
Title	Create
Description	The system shall allow the users to create a new PIR.

Rationale	To allow users to create a new PIR.
RRI	R1
Priority	1

ID	SR2
Title	Categorize
Description	<p>The system shall allow the users to select the category of PIR upon creation of a new PIR.</p> <p>Once users change the category of the existing PIR, the content within it will be replaced with blank values.</p>
Rationale	To allow users to select the category of the PIR.
RRI	R3, R4, R5
Priority	1

ID	SR3
Title	Update
Description	<p>The system shall allow the user to update the content for a specific PIR from the one he/she has created.</p> <p>Once the PIR has been updated, the system shall be able to display the latest content in the PIR.</p>
Rationale	To allow user to update the content in the created PIR.
RRI	R6
Priority	1

ID	SR4
Title	Search
Description	<p>The system shall allow the user to search for a specific PIR from the one he/she has created.</p> <p>The system should allow users to search for a specific PIR based on UID or search criteria.</p>

	<p>On successful retrieval, the system shall show the search results to the user.</p> <p>If no PIR is found, the system shall return an appropriate message to notify the user that no record was found for the search.</p>
Rationale	To allow users to easily retrieve a specific PIR without going through the whole list of records.
RRI	R7
Priority	1

ID	SR5
Title	View
Description	The system shall allow users to view the content of PIRs once selected by users.
Rationale	To allow users to view the content within the created PIRs.
RRI	R8, R15
Priority	2

ID	SR6
Title	Delete
Description	<p>The system shall allow users to be able to delete any existing PIRs.</p> <p>Users are not able to retrieve the deleted PIRs.</p>
Rationale	To allow users to remove the existing PIR.
Reference	R9
Priority	1
ID	SR7
Title	Save
Description	The system shall allow users to save the created PIR.
Rationale	To allow users to store PIR in local storage.
RRI	R10
Priority	2

ID	SR8
----	-----

Title	Load
Description	<p>The system shall be able to load and open the PIR only with the appropriate file extension.</p> <p>The system shall output the content in the opened PIR to the user.</p>
Rationale	To allow users to open and load content in previously saved PIR.
RRI	R11
Priority	2

ID	SR9
Title	Check ID
Description	<p>The system shall be able to check the existence of PIRs with UID.</p> <p>If a record with matched UID is found, the system will conduct appropriate handling according to the function.</p> <p>Otherwise, the system will output an error message to the users.</p>
Rationale	To allow the system to check the existence of PIR for validation purposes.
RRI	R13, R14
Priority	2

***Events**

ID	SR10
Title	Update Alarm
Description	The system shall be able to delete or update an event's alarm.
Rationale	This allowed the user to manage the alarm information of an event.
RRI	R6, R16

Priority	2
----------	---

Non-functional Requirements

Speed

- The system shall be operational once required usage by users, and the response time for the system towards users' operations should be less than 1 second.

Reliability

- The system shall provide appropriate responses according to users' selection, and the system shall perform without failure 99 percent of the time. The system shall display an error message to the user if the requested record does not exist.

Robustness

- Once failure occurs, the system should be recovered within 2 minutes.