Software Requirements Specifications

Personal Information Manager

COMP3211 SOFTWARE ENGINEERING

PROJECT GROUP 19

Name:	SID:
Cheuk Hang CHEUNG	21032932D
Chun Ming FONG	21019558D
Ho Nam FAN	22110264D
Wai Tak WONG	21108992D

Content

1 Preface

Expected Readership: Personal information manager users

Version: beta 1.0

Update: N/A

Update rational: N/A

2 Introduction

2.1 Purpose

To create a Personal Information Manager which could be operated to store user personal information.

2.2 Scope

- Provide users with an efficient centralized system to store and manage their personal information.
- Users shall be able to manage their information easily and effectively
- Our system is provided to the user group who want to mange their personal information and organize them in an efficient manner. We provide the function for users to manage and store their contacts, events, notes, and tasks.

2.3 Intended audience

This project is a prototype for a personal information manager. This project is useful for people who want to have a simple and efficient management system for storing their personal information.

2.4 Document Conventions

This document uses the following conventions.

Abbreviation	Meaning
CLI	Command Line Interface
MVC	Model-view-Controller
NFR	Non-Functional Requirement
PIM	Personal Information Manager

PIR	Personal Information Record
SR	System Requirement
UR	User Requirement

3 User Requirements Definition

3.1 User Functional Requirements

- **UR-01:** The user can create different type of personal information records (PIRs) in the PIM
- **UR-02**: The user can create new plain text PIRs for taking quick notes
- **UR-03**: The user can create new tasks PIRs with descriptions and deadlines.
- **UR-04**: The user can create new event PIRs with descriptions, starting times, and alarms.
- **UR-05**: The user can create contact PIRs with names, addresses, and mobile number.
- **UR-06**: The user can modify the data in existing PIRs
- **UR-07**: The user can search based on criteria concerning their types and the data stored in their fields.
- **UR-08**: The user can search is a given string is inside the stored information.
- **UR-09**: The user can search by the time (deadline, start time) with before (<), after (>), or equal to (=) a given time
- **UR-10**: The user can combine search with logical connectors and (&&), or (||), and negation (!)
- **UR-10**: The user can print out detailed information about the PIRs.
- **UR-11**: The user can delete a specified PIR.
- **UR-12**: The users can store PIRs in a file with the extension name ".pim".
- **UR-13**: The users can load PIRs from a file with the extension name ".pim".

3.2 User Non-Functional Requirements

- **UR-14**: The Personal Information Manager runs as a command-line system.
- **UR-15**: The user shall not enter PIR with empty information.
- **UR-16**: The user should enter the information in the correct format.
- **UR-17**: The User should manually use the store function if they want to store the information for long-term storage
- **UR-18**: The information in the PIM should only be available to the user who wrote the records.

4 System Architecture

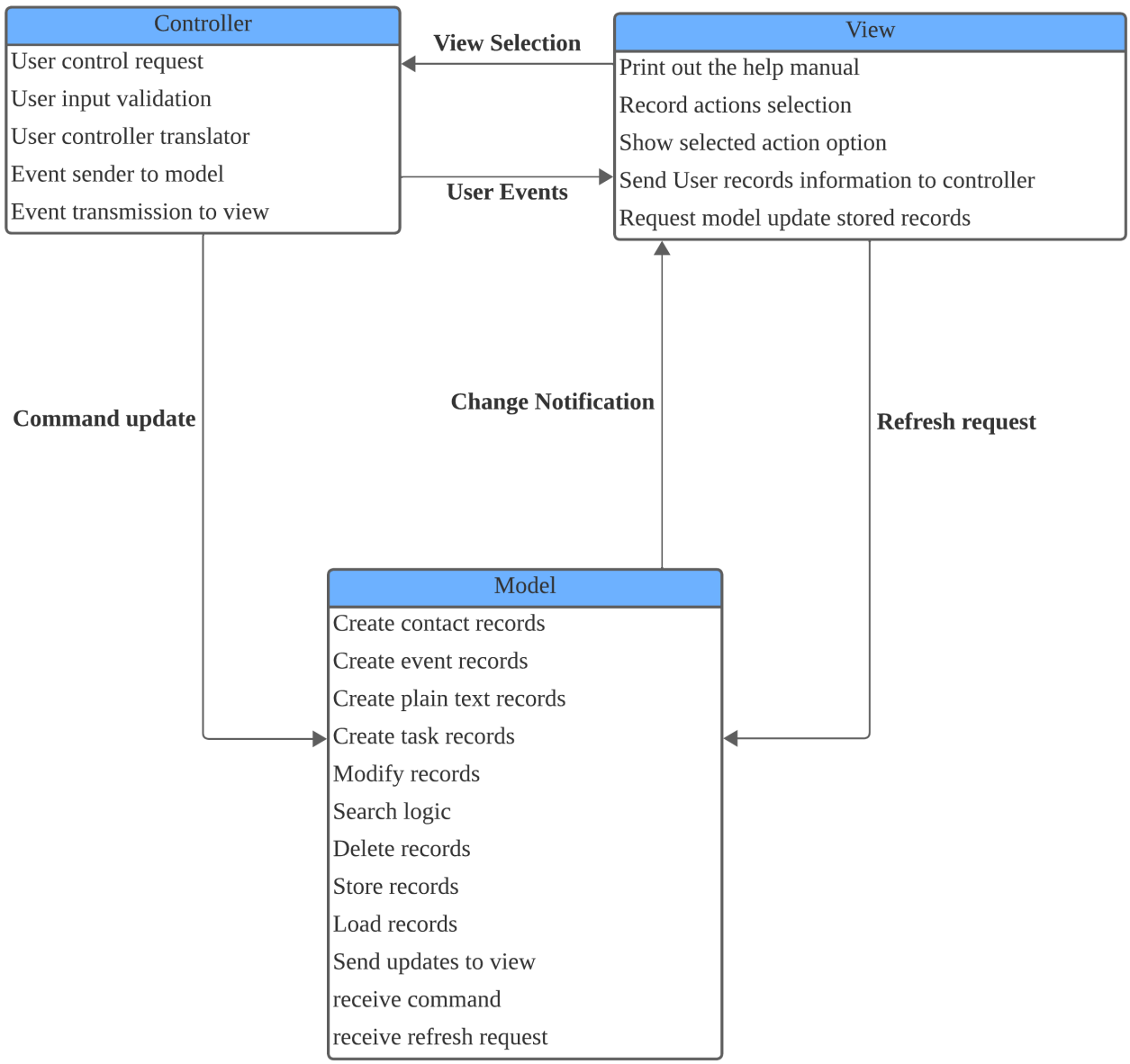
4.1 Architectural Patterns

The **Model-View-Controller** (MVC) separates the presentation from the interaction of the system data. The system is divided into three mutually logical building blocks. The **Model** component building block manages all data and accepts actions that transmit user commands. The **View** component defines the user interface that is presented to the user. The **controller** component manages the user actions, such as user selecting to create a record, and these actions are transmitted to the view model

4.2 Reasons and Requirements

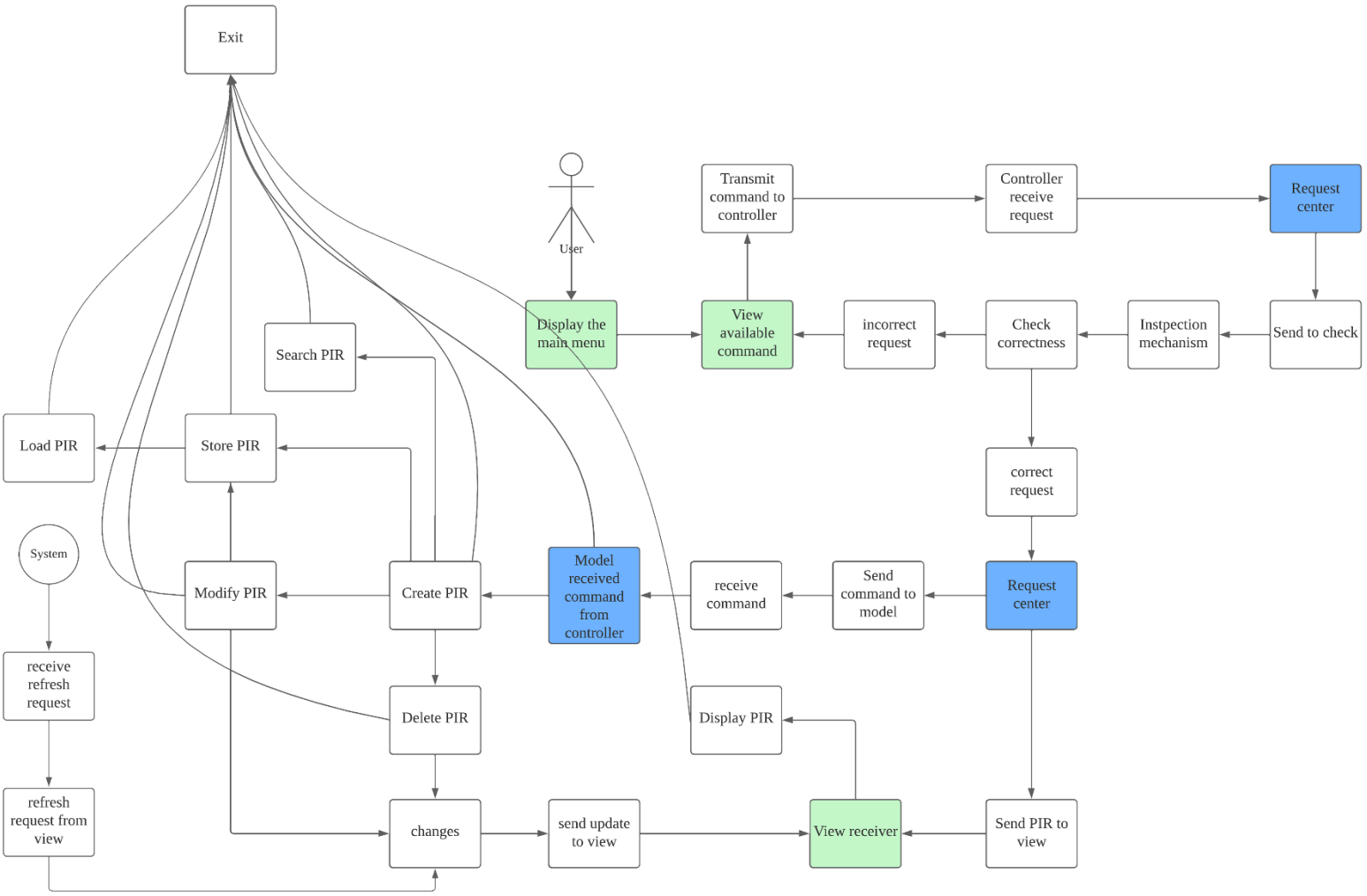
1. The MVC pattern allows the system to be separated into 3 parts. Each part is responsible for different and exclusive tasks with little overlap. Result in a good encapsulation of each functional part, and enhances their cohesion, which also benefits the API design of the system.
2. The MVC pattern allows the system to increase reusability of the functions. This allows modification to one component without affecting the other components.
3. The MVC pattern allows the system to interact with the data in vary ways, this allows simple changes or addition of new presentation ways of the data.

4.3 MVC Architecture Patterns



The overall structure is divided into three parts including *model, view, and controller.* The controller part mainly receives the users' action with a translator to turn it into model language and sends an update to the model for further logical operation and return notification to view users' interaction. on. The view part mainly contains users' actions and visualization contents with CLI for users to see what action to take. The model part contains the creation, modification, deletion of all the records, search logic, save and l and loading file of the PIM.

4.4 activity diagram



5 System Requirements Specification

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

5.1 System Functional Requirements

ID	SR-01
Title	Creation of records of contacts, events, notes, and task
Requirement	System shall provide the functionality to create records of contacts, events, plaintext, and task.
Rationale	Allow user to store their personal information
Reference (SR, UR)	UR-01, UR-02, UR-03, UR-04, UR-05
Priority	1

ID	SR-02
Title	Modification of records of contacts, events, notes, and task
Requirement	System shall provide the functionality to modify the stored records of contacts, events, notes, and tasks
Rationale	Allow user to modify their existing personal information
Reference (SR, UR)	UR-06
Priority	1

ID	SR-03
Title	Search on records
Requirement	System shall provide the functionality to search stored records by searching is the given string is inside the stored records, or to search by before, after, or equal to a given time.
Rationale	Allow user to search their existing personal information
Reference (SR, UR)	UR-08, UR-09
Priority	1

ID	SR-04
Title	Combine search on records
Requirement	System shall provide the functionality to combine search with the operator and (&&), or (||), and negation (!).
Rationale	Allow user to more effectively search their existing personal information
Reference (SR, UR)	UR-10
Priority	1

ID	SR-05
Title	Display all stored records
Requirement	System shall provide the functionality to display all stored records
Rationale	Allow user to see all their existing personal information
Reference (SR, UR)	UR-10

Priority	1

ID	SR-06
Title	Deletion of records
Requirement	System shall provide the functionality to delete existing records
Rationale	Allow users to delete their existing personal information as they may no longer need those records for reminder.
Reference (SR, UR)	UR-11
Priority	1

ID	SR-07
Title	Long-term storage of records
Requirement	System shall provide the functionality to store user input records into a ".pim" file for long-term storage.
Rationale	Allow users to save their information and can be used for future retrieval.
Reference (SR, UR)	UR-12, UR-17
Priority	1

ID	SR-07
Title	Loading records from file
Requirement	System shall provide the functionality to load user stored information from file ".pim"
Rationale	Allow user to retrieve their information that are stored in the given formatted file
Reference (SR, UR)	UR-12, UR-17
Priority	1

5.2 System Non-functional Requirements

ID	SR-08
Title	Command-Line user interface
Requirement	The Personal Information Manager (PIM) operates as a command-line system and provides a text-based interface for user interaction.
Rationale	The command-line interface may enable lightweight and efficient operation.
Reference (SR, UR)	UR-14

ID	SR-09
Title	Mandatory Field Validation
Requirement	The system provides a mandatory field validation of Personal Information Records (PIRs) to prevent the user input the empty information to the system.
Rationale	This function ensures the data completeness and correctness, avoiding discrepancies and delivering relevant information.
Reference (SR, UR)	UR-15

ID	SR-010
Title	Data Format Validation
Requirement	The system evaluates the format of information submitted by the user in PIRs to ensure that it follows the specified standards for each entry.
Rationale	A specified data format can help maintain data integrity and prevent errors or incorrect data entry.
Reference (SR, UR)	UR-16

ID	SR-011
Title	Manual Storage Functionality
Requirement	The system requires the users to turn on the “store” function to save the information for long-term storage.
Rationale	The requirement for explicit user action for storage provides a choice to users which information is kept beyond the current session.
Reference (SR, UR)	UR-17

ID	SR-012
Title	User specific data access
Requirement	The Personal Information Management System (PIMs) ensures that only the user who created the record has access to the personal information stored in the system.
Rationale	The restrictive access to specific user records enhances data security and privacy.
Reference (SR, UR)	UR-18