

COIT20256/COIT23001 Assignment 2

Assessment item 2 — OOP and Data Structure

Due date: Thursday of Week 11

ASSESSMENT

Weighting: 30%

2

Objectives

- Demonstrate an understanding of Object-Oriented Programming concepts in Java.
- Gain practical skills in Graphical User Interface (GUI) programming by implementing an event-driven interface.
- Develop and test stand alone Java applications.
- Evaluate algorithm, data structure and program designs used in developing Java applications.

Assessment Task

Specification for the Queensland Children Allergy Information System

Introduction

For the second assignment, you will develop a windowed application to assist the research staff of a medical centre to maintain and manage a list of personal data related to Queensland children with allergies.

Overview of the application

In a real-world software application, you would ideally use a database to store patient information. In this application however, you will use a text file to store relevant information. For simplicity, only 5 attributes of a patient will be stored: patient name, telephone, age, gender and allergy source. The system allows the data entry in such way that the patient can have the same name but different phone number. If this occurs, they will be treated as different data entries.

In a typical session of the application, the patient data file will be loaded from the disk and displayed. The research staff then can view, sort, search, as well as add/delete data entry. The added information can be saved back to the data file. This application will provide a comprehensive exercise in text file processing, sorting, searching, and using data structures. As well, you will create a new GUI with various components.

The GUI

The GUI you will create should resemble the screen shown below.

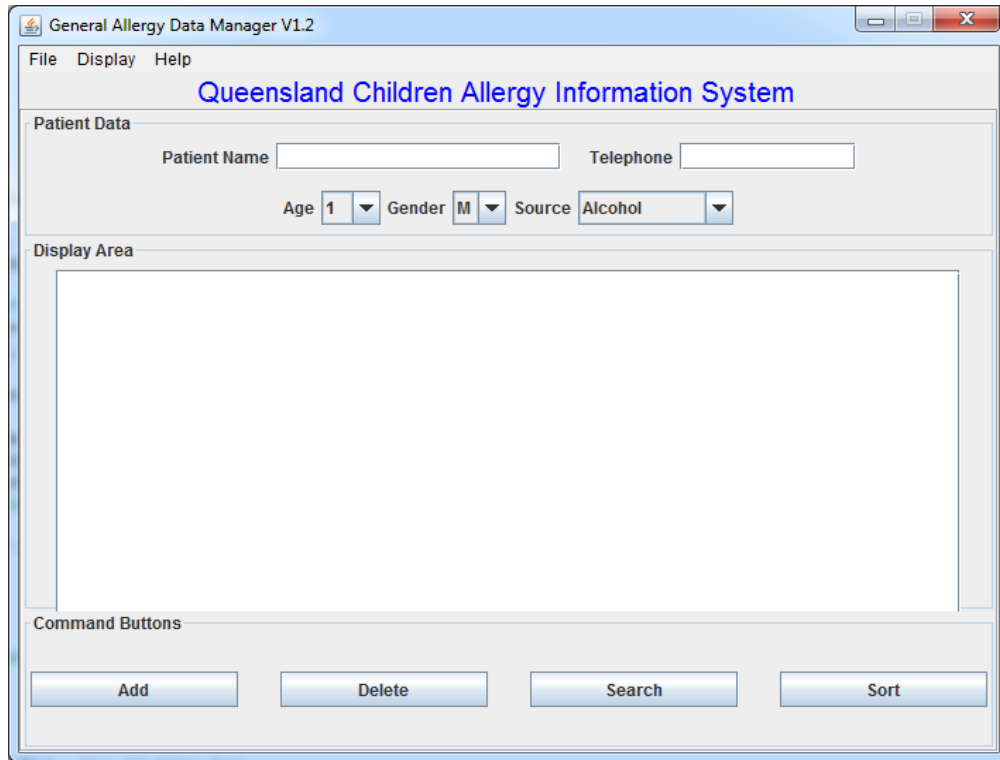


Figure 1

The GUI components on the screen as shown in Figure 1 include:

- A label for the application title – Queensland Children Allergy Information System
- A panel which contains the rest of the components (on sub-Panels) on the screen namely:
 - o A sub-panel containing 2 text fields and their corresponding labels, 3 JComboBox and their corresponding labels;
 - o A sub-panel containing a text area for displaying the patient data; and
 - o A sub-panel containing 4 buttons.

In addition, the GUI also contains a few of menu items, three under **File** menu, two under **Display** menu and one under **Help** menu. When the user clicks the **File→ Load** menu item, the program will load the data from a data file to the system. The function specifications of other menu items can be found on page 8. Overall these menu items are:

File->Load **Display->Show** **Help->About**
->Save ->Clear
->Exit

Some details of how the program functions

1. In a typical session of the application, when the program runs, normally the first action is to load data from a data file by selecting the file menu item: **File->Load** so that the relevant data in the data file is read and the content of loaded file is displayed on the text area. Below is the typical screen when the sample data file is first loaded. Note that if the data were not loaded but the 'Sort' button is pressed, a message like "Data to be loaded" should be displayed on the text area.

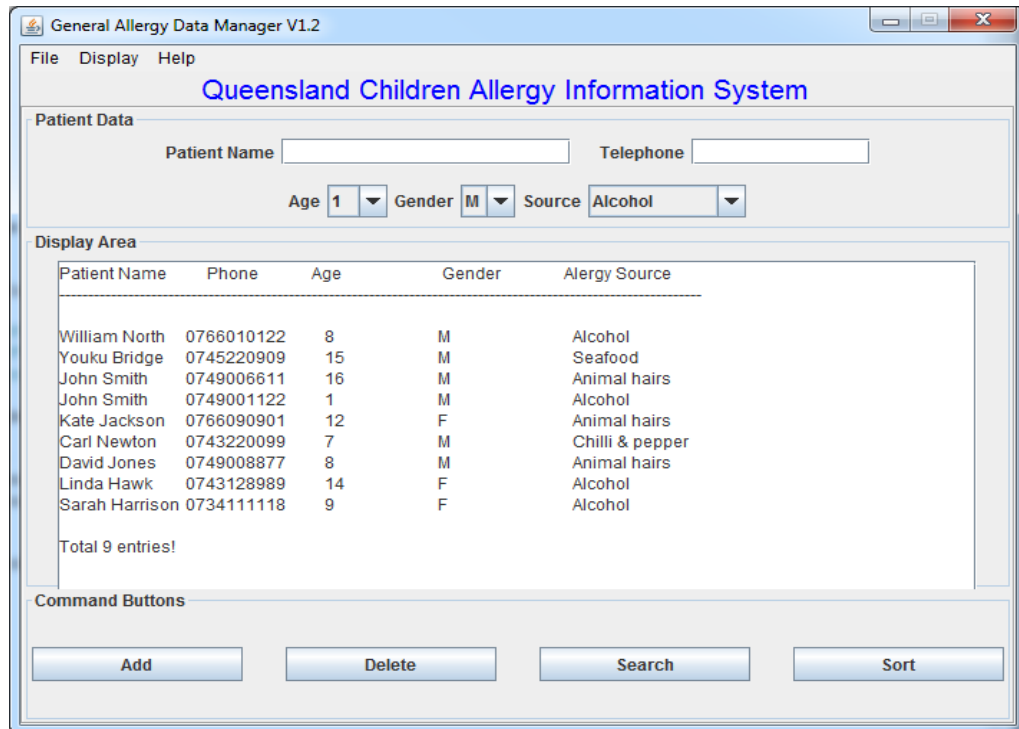


Figure 2

2. **Add.** Now switching to the button operations, here is the screen when the information for a new patient (*Tony Stone* with relevant information) is added to the list and displayed as shown in Figure 3a (before the Add button is pressed.) and Figure 3b (After the 'Add' button is clicked, where the text fields and JCombobox are required to return to the initial default status.).

General Allergy Data Manager V1.2

File Display Help

Queensland Children Allergy Information System

Patient Data

Patient Name: Telephone:

Age: Gender: Source:

Display Area

Patient Name	Phone	Age	Gender	Allergy Source
William North	0766010122	8	M	Alcohol
Youku Bridge	0745220909	15	M	Seafood
John Smith	0749006611	16	M	Animal hairs
John Smith	0749001122	1	M	Alcohol
Kate Jackson	0766090901	12	F	Animal hairs
Carl Newton	0743220099	7	M	Chilli & pepper
David Jones	0749008877	8	M	Animal hairs
Linda Hawk	0743128989	14	F	Alcohol
Sarah Harrison	0734111118	9	F	Alcohol

Total 9 entries!

Command Buttons

Add Delete Search Sort

Figure 3a

General Allergy Data Manager V1.2

File Display Help

Queensland Children Allergy Information System

Patient Data

Patient Name: Telephone:

Age: Gender: Source:

Display Area

Patient Name	Phone	Age	Gender	Allergy Source
William North	0766010122	8	M	Alcohol
Youku Bridge	0745220909	15	M	Seafood
John Smith	0749006611	16	M	Animal hairs
John Smith	0749001122	1	M	Alcohol
Kate Jackson	0766090901	12	F	Animal hairs
Carl Newton	0743220099	7	M	Chilli & pepper
David Jones	0749008877	8	M	Animal hairs
Linda Hawk	0743128989	14	F	Alcohol
Sarah Harrison	0734111118	9	F	Alcohol
Tony Stone	0748110099	1	M	Alcohol

Total 10 entries!

Command Buttons

Add Delete Search Sort

Figure 3b

Note: 1) if the input data entry -with identical patient name and phone is existed in the system, the program will show a warning message –“Data entry existed! Re-input!” on the text area, subsequently the data entry is not added into the system in this case; 2) if the name text field or phone text field is not entered any data and the Add button is clicked, then a message box will pop out as below.

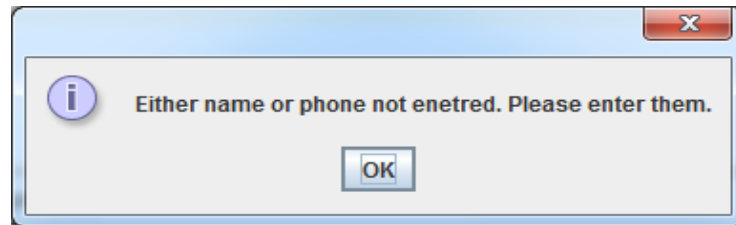


Figure 3c

3. **Delete.** The ‘Delete’ button requires to enter a patient name and corresponding phone to be deleted from the list via the name and phone text fields as shown in Figure 4a with an example. After clicking the ‘Delete’ button, the patient data (if existed) is removed and a confirmation message as Figure 4b will pop out subsequently. If the entered data entry is not existed in the list, a message box with the content “Patient xxxx (name) not found” pops out as shown in Figure4c. If the name or phone text filed is not entered any data and the delete button is clicked, a message box will pop out as Figure 4d.

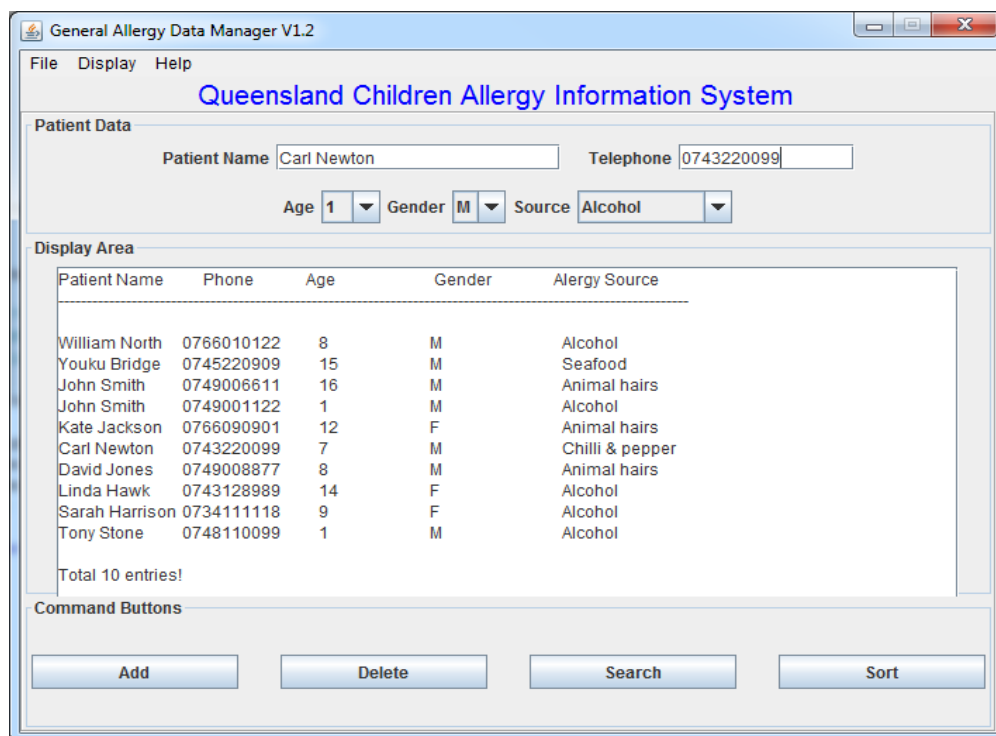


Figure 4a



Figure 4b



Figure 4c



Figure 4d

- 4. Search.** The 'Search' button requires to enter a patient name regardless of lower case or upper case spelling via the Patient name text field. If the patient name exists in the system, the corresponding information of this patient is display on the text area as shown in Figure 5; otherwise the message – “No result found for the patient: xxxx (*the input name*)” is displayed on the text area.

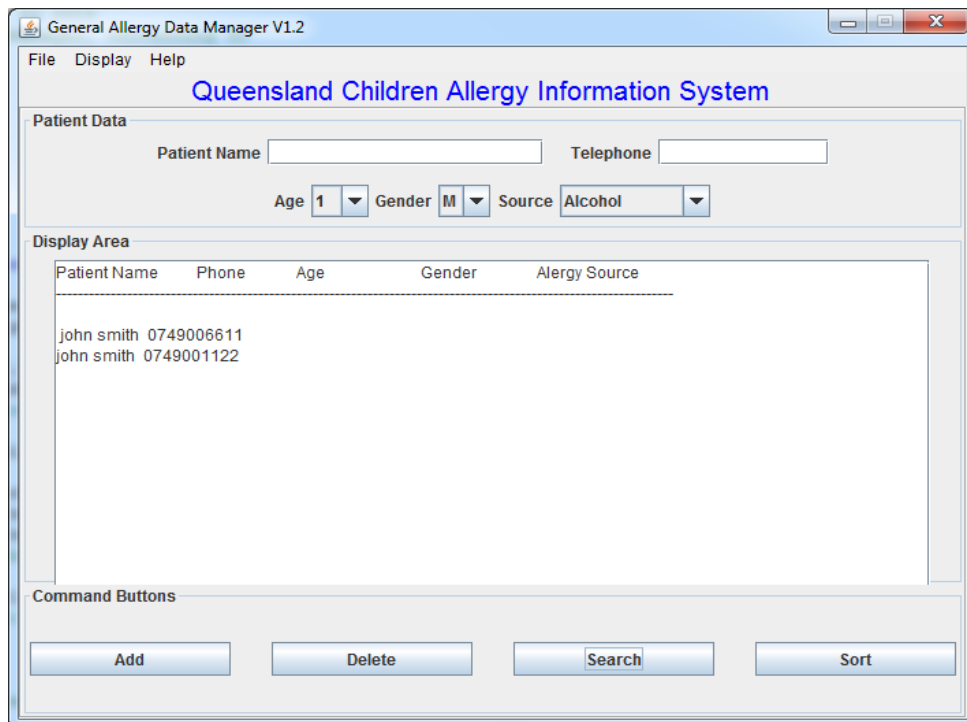


Figure 5

5. **Sort.** Here is the screen on which the list is sorted by name when the 'Sort' button is pressed. You can use any of the sorting algorithms studied in the textbook to sort the list. We would like to recommend you using the patient class with implementation of *Comparable* interface so that two patient objects can be easily compared by their names. Figures 6a and 6b are the screen shots before sorting and after sorting respectively.

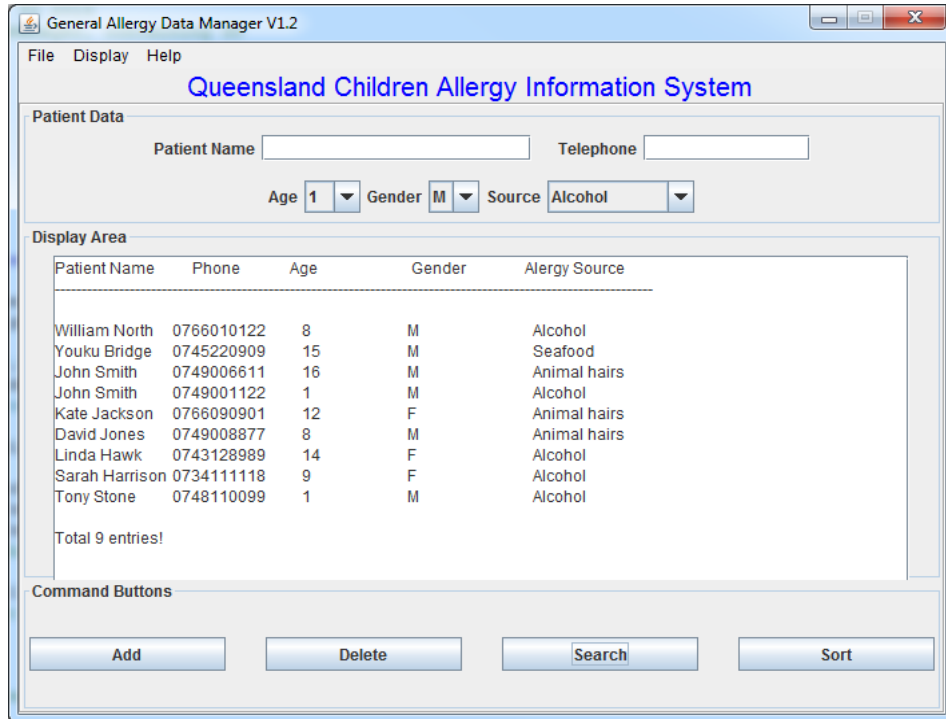


Figure 6a

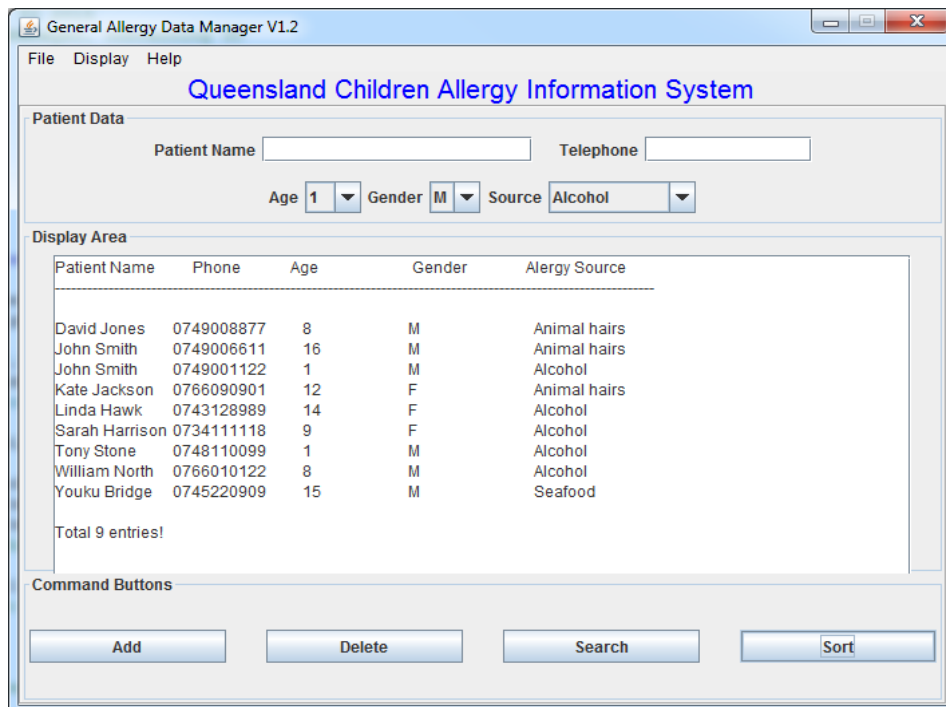
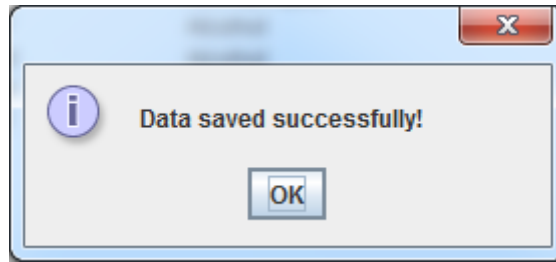


Figure 6b

6. **File->Save.** This menu option will save the updated list to the data file. A confirmation message should be given to the user if the old file is successfully overwritten with the updated one.



File->Exit

The system will exit its execution if this menu operation is selected.

7. **Display->Show**

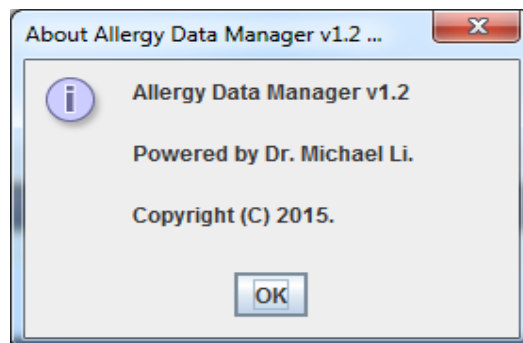
The application will display all data to the text area when this menu is selected.

Display->Clear

The application will clear all data on the text area and re-set text fields and combo box to default status.

8. **Help->About**

The system will pop out an information message box (as shown below) if this menu operation is selected.



9. **Important:** Structure your program by using helper methods. The main technique is to identify the main tasks and then break these down to small manageable tasks that can be translated easily into suitable named methods.

Data storage structures

1. A text file called Patient.txt will be used to store the patient information. This file will be made available to you on the course website.
2. The patient list is loaded from the above file into a LinkedList or ArrayList of Patient objects. For this you will need the Patient class file - Patient.java. For convenient comparison of Patient objects, we suggest you to implement *Comparable* interface based on comparison of the patient name for this class.
3. Use meaningful variable names to store all values required in the application

Implementation platform

You will implement your program in Java using either the TextPad Editor (strongly recommended) or NetBeans. NetBeans version 6.9 or later is available in University computing labs but may also be downloaded from the following site: <http://netbeans.org/downloads/index.html>

TextPad is also provided in the University computing labs. It can also be downloaded from <http://textpad.com/download/index.html>

Assignment submission

1. You must submit your assignment using the Moodle online submission system.
2. You will submit three (3) files:
 - a) The source code of your program (file name)-PatientManager.java
 - b) the Patient.java file
 - c) the Patient.txt file (Although this text file is available from the course website).

You must zip up your files and name the zip file Assignment2.zip before submitting. Do NOT submit an executable (.class) version of your program, it will not be marked.

3. **IMPORTANT:** You must make a back up of your assignment before submitting in case your submitted copy becomes corrupted.

Assignment 2 Marking criteria

Item	Marks
GUI overall presentation	5
File menu	
Load	3
Save	3
Exit	0.5
File menu	
Show	1.5
Clear	0.5
Buttons	
Add	4
Delete	4
Search	2.5
Sort	2
Patient class implementation	2
Quality of code (comments, indentation, naming, and readability etc)	2
Total	30
Deduction (Plagiarism, late etc)	
Final total	

Note: • If your program doesn't compile or run, partial marks will be allocated by inspection of the source code.

