

**INDIVIDUAL PROJECT 2**

**SEMESTER 2 2019/2020**

|  |  |
| --- | --- |
| **COURSE CODE** | **:** **KK14203** |
| **COURSE NAME** | **: OBJECT-ORIENTED PROGRAMMING** |
| **TITLE** | **: VETERINARY REGISTRATION SYSTEM** |
| **NAME** | **: HAZMI BIN HASHIM** |
| **STUDENT ID** | **: BI19110034** |
| **SECTION** | **: 1** |
| **LECTURER** | **: DR. MOHD SHAMRIE SAININ** |

**Object Oriented Concept Implementation**

1. Object & Classes

Some examples of the classes that being used in the program including:

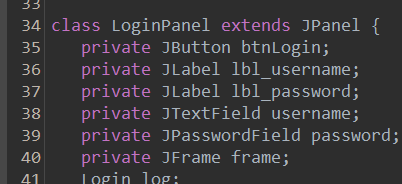
MainPanel, LoginPanel, registerPanel, historyPanel

They have the object that can be created outside the class to inherit the data. The “Frame” classes build the frames by adding the panels. Meanwhile, the JPanel components such as buttons and text fields being builded by the “Panel” classes.

The main() method is found in the class called “StartLogin”, and contains the following code to start the program: new LoginPanel();

1. Encapsulation

For example, the private variable at line 34 until line 40, means that the variable only can be accessed in the LoginPanel class and can not be accessed outside of the class.



1. Exceptions

The try statement allows you to define a block of code to be tested for errors while it is being executed. It’s a little bit similar to a loop function, while executing, do test error. Meanwhile, if there is an error occurs in the try block, the catch statement can allows you to define a block of code to be executed.

For example, the program have try and catch statements from line 560 to line 574.

1. Inheritance

A class can act as a java swing by itself, for example, where they inherit the javax.swing properties by using extends.

In the program, we have line 90 class MainPanel extends JPanel and line 140, class registerPanel extends JPanel.

1. Polymorphism

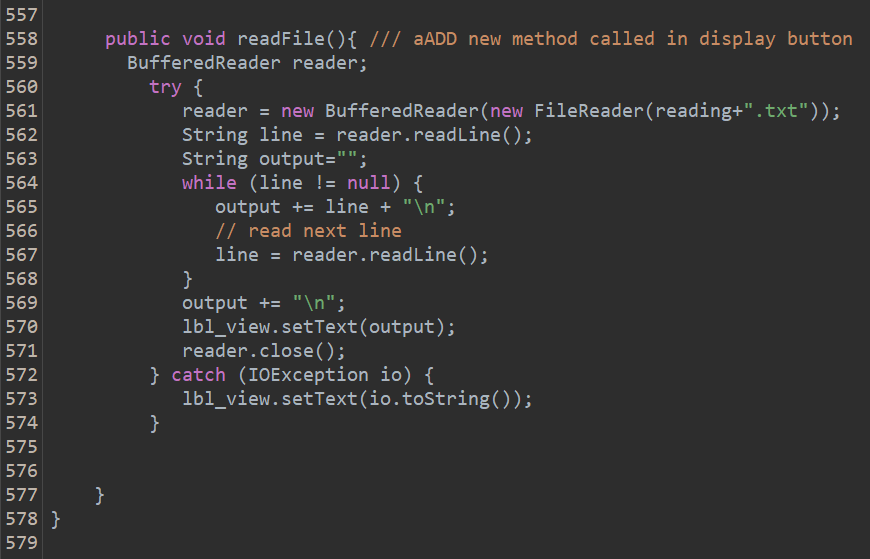
Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.

For instance, line 424 and line 436 have the same lbl\_output.setText(output); but both does not interfering with one another.

**Read and Write Implementation**

1. Read

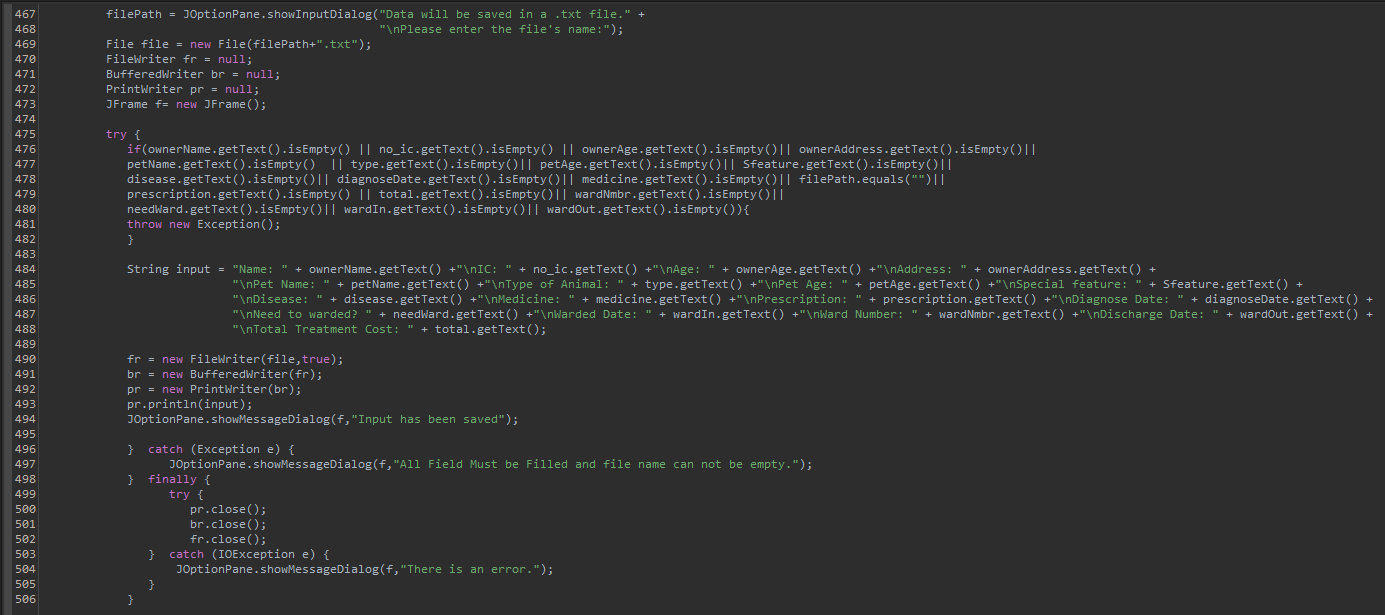
Using the program, the user can make the program can read the saved file (that previously saved by the user) and display it. Below is one of the snippets of where the program used it.

****

If the user type a file name, and a file with the exact same name do exist inside the folder, then the program will display the file’s data to the user. (Refer User Manual, Step 7 and 8). Meanwhile, if there aren’t any file with that name, the program will display this: “java.io.FileNotFoundException: ayam.txt (The system cannot find the file specified)”. That means, there are no such file in the folder with the name that the user search for.

1. Write

For write implementation, the program will save the data that have been saved by the user and keep it in a specific text file (.txt). Below the example of the snippets from the program. It starts at line 466 until line 508.



If the user has done fill in all the field, they can save all the data in one file. When the “Keep Record” button being selected, the user will be required to name the file. Then a text file with that name will be created in the “OOP Project 2” folder. (Refer User Manual, Step 5)

**User Manual**

1. First and foremost, the user need to run the StartLogin.java file. When the user enter their respective username and password (based on Figure 1), and press the login button, they will be welcomed by a message (thus marking it as a successful login) as in Figure 2. For example, fill in the required field with ‘username’ and ‘password’ respectively.

\*Fill in Username > Password > Login > OK >> User logged in.

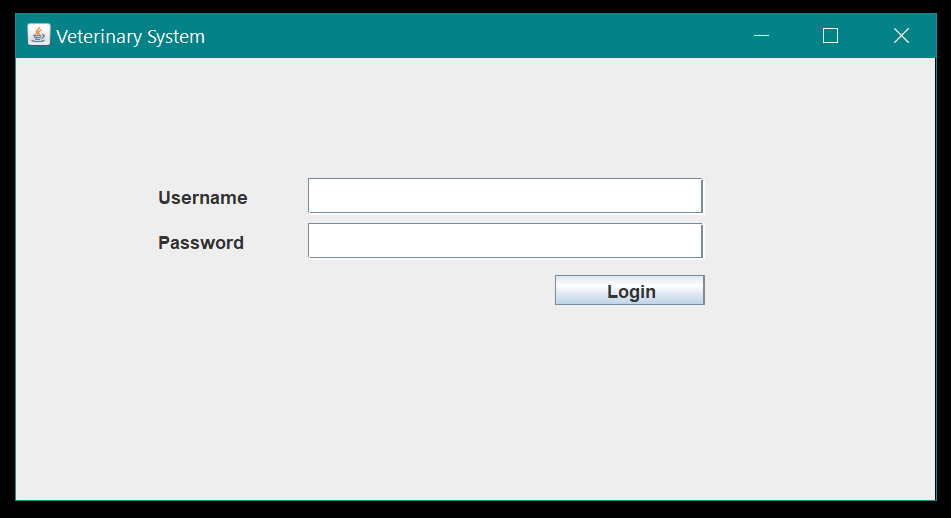


Figure 1

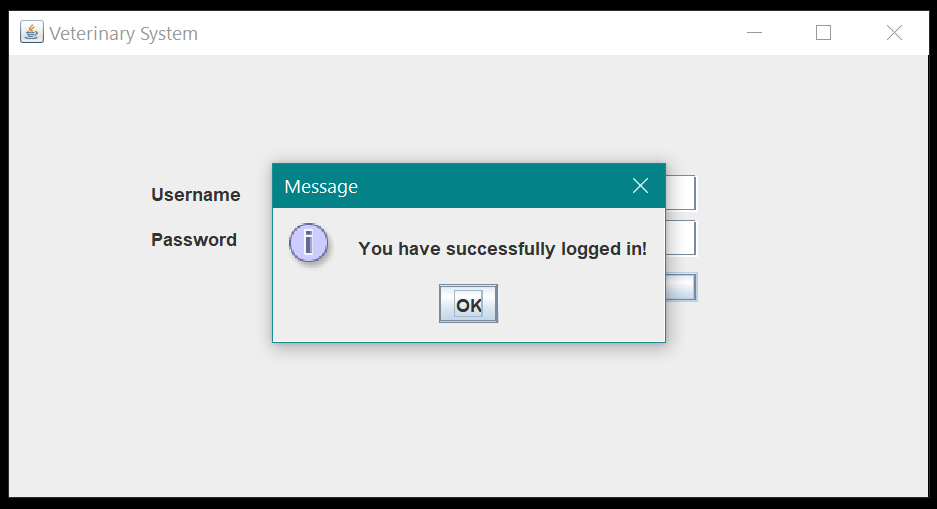


Figure 2

1. When the user selected the “OK” button, the system later will display a new window, as Figure 3 shows.

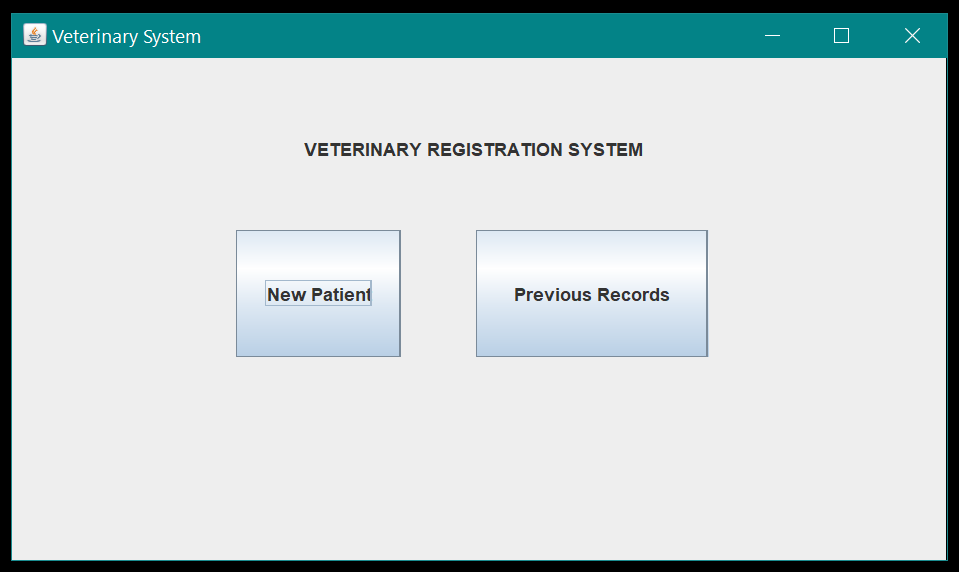


Figure 3

1. When the user select the first button, the “New Patient” button, they will be directed to a new window. The user need to fill in all the needed details in each specific tab, such as the owner profile, their pet details, diagnose details and also ward details as shown by Figure 4 and Figure 5.

\*New Patient > Owner Details > Pet Details > Diagnosis > Ward

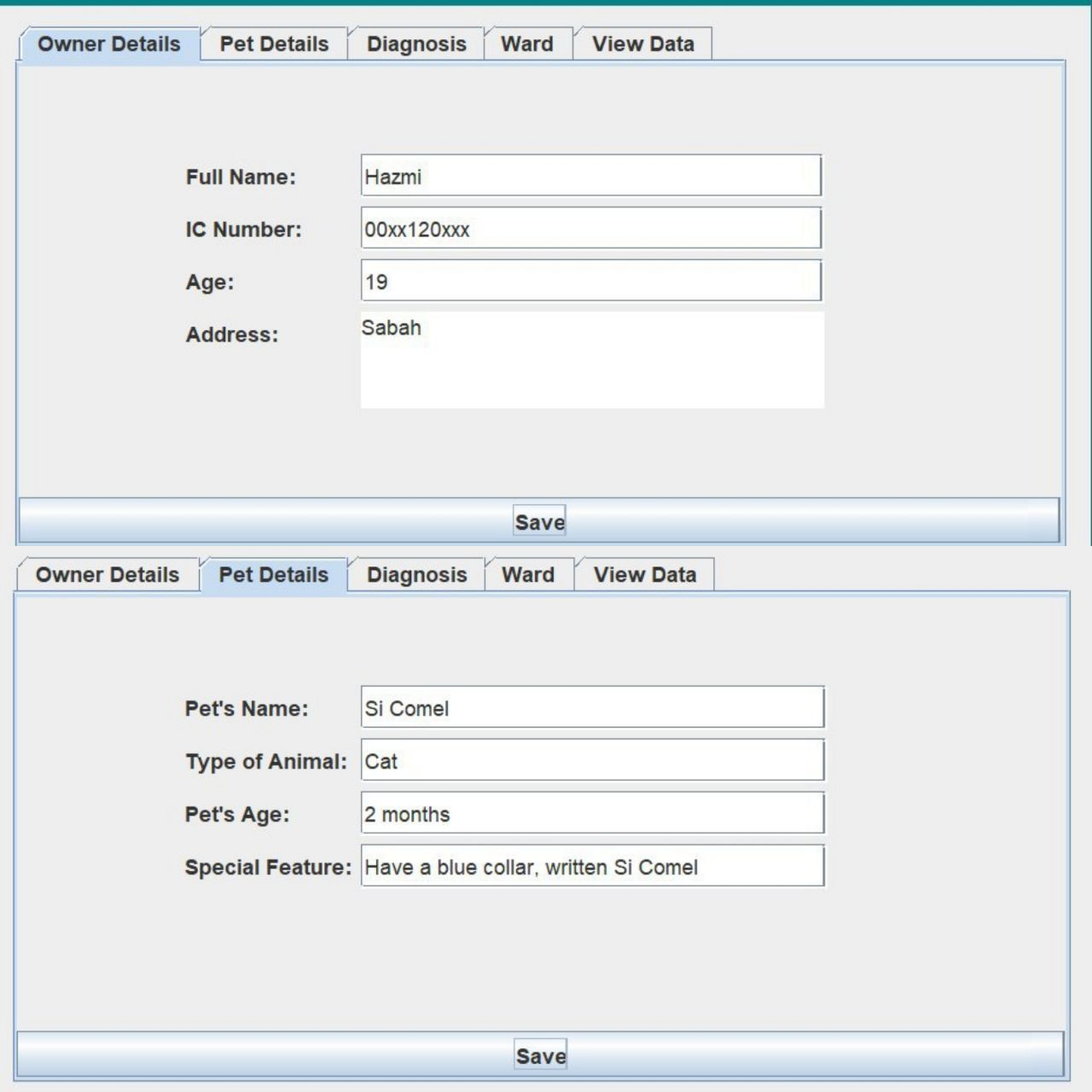
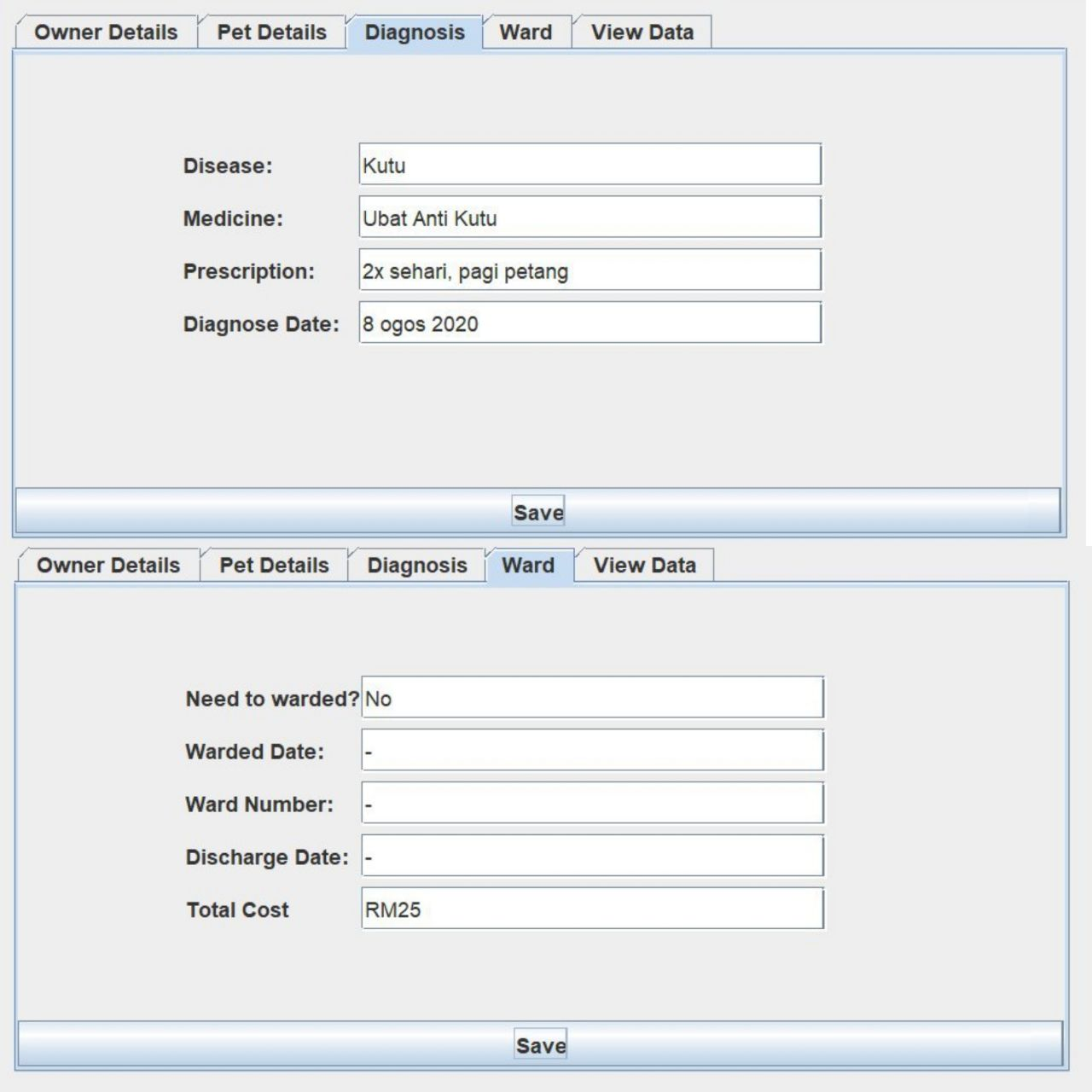
 

Figure 4 & Figure 5

1. If the user want to know the data that they have previously saved (from the previous tab), they can look it at the last tab, the “View Data” tab. For example, when the user has saved the pet details data from the second tab, if they click the View Data tab right away, they will be shown the saved data, as shown at the Figure 6.

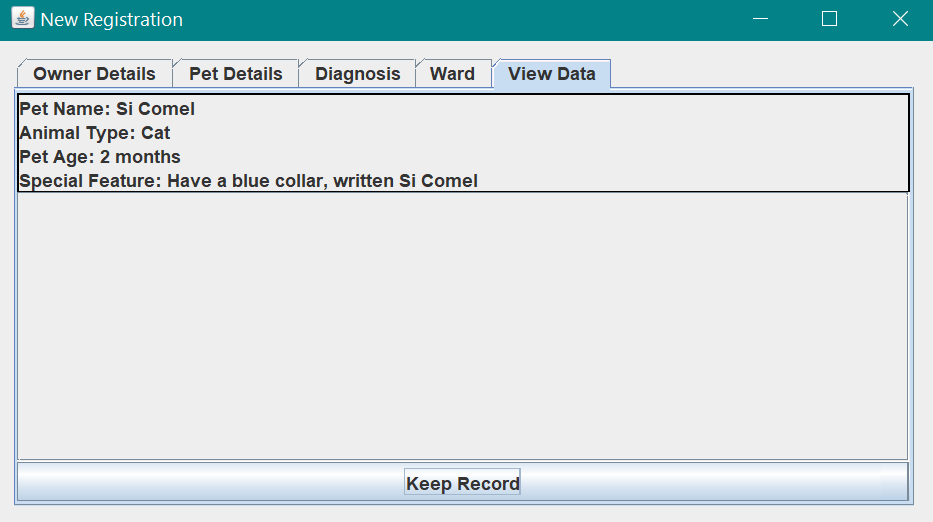


Figure 6

1. When the user has done fill in all the data, they can save it in a text file (.txt) by clicking the “View Data” tab, and click the “Keep Record” button at the bottom. The user then will be asked to give a name the file (my recommendation is to put both owner and the pet name together) before it automatically saved.

\*View Data > Keep Record > Naming the File > OK >> File Saved.

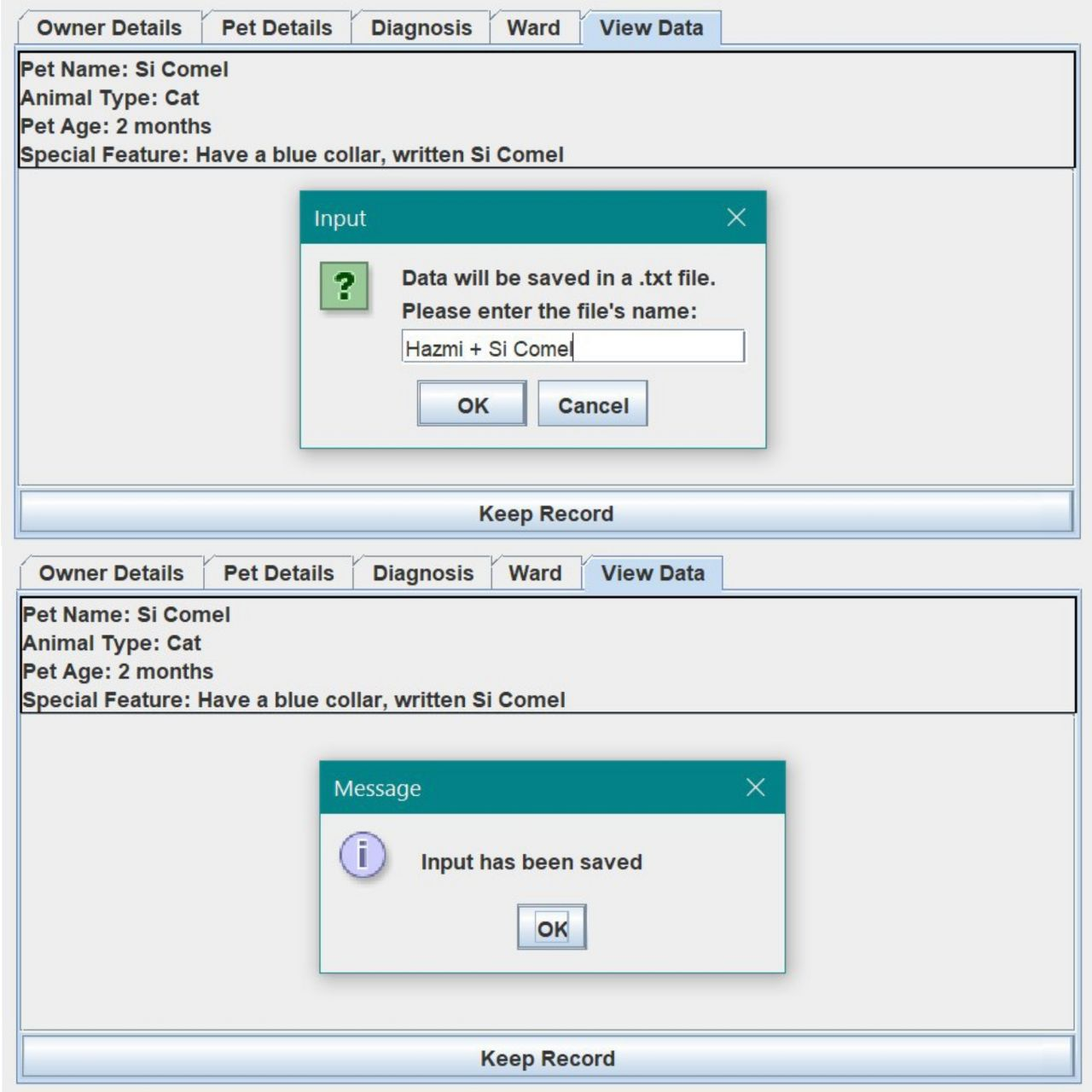


Figure 7

1. There are two possible ways to search the previous saved files. The **first** one is by searching the files manually through the same big folder, “OOP Project 2” (as Figure 8 shows).



Figure 8

1. The **second** way is by clicking the “Previous Records” button at the main system panel (refer Step 2, Figure 3). The user then will be directed to a new window. By selecting the “Display” button (at the bottom-center), the user need to enter their desired file’s name. For example, Testing1.

\*Previous Records > Display > File’s Name > OK >> File’s Data Displayed.

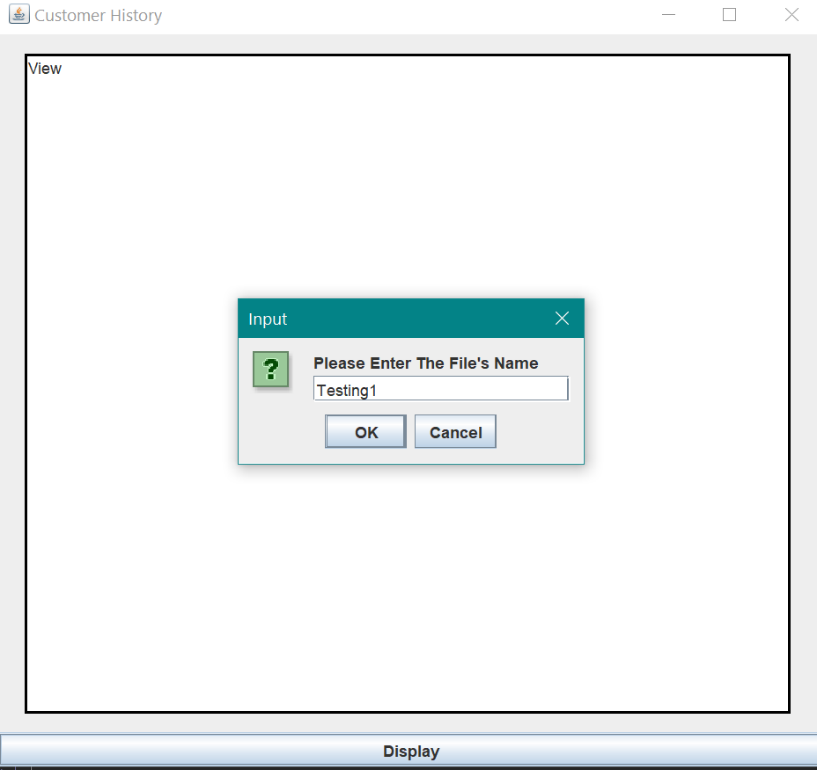


Figure 9

1. If the file is available in the folder, the user will be displayed with the file’s data. Refer the Figure 10 below, by using the Testing1 file.

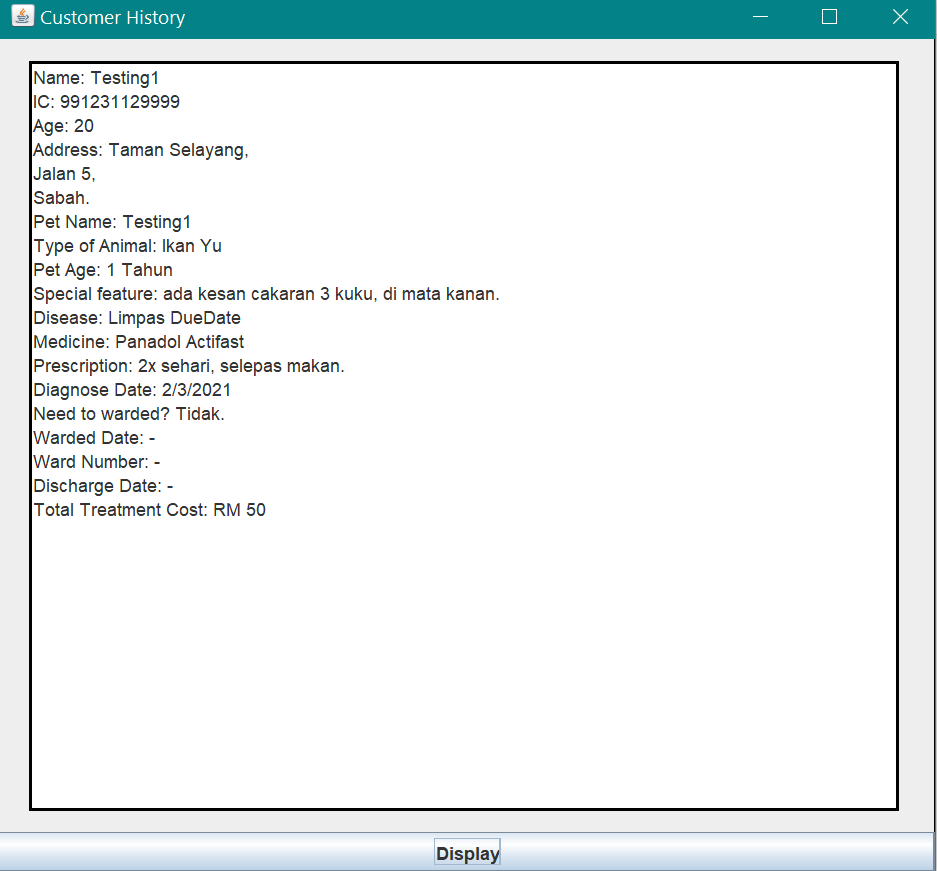


Figure 10

1. For a reminder, selecting the close button (at the top right window panel, symbolize with X) during the “New Registration” panel (refer Step 3, Figure 4 and 5), the one in tabbed panel layout, that will resulting user to go back to the main system panel. Same with the “Customer History” panel (as Figure 10), the user will be directed back to the “Veterinary System” panel if they close the window panel. Meanwhile, selecting the close button at the main panel system will end the system program.