# C:\Users\Asus\Documents\download.jfifC:\Users\Asus\Documents\download (1).jfifhttp://www.ums.edu.my/ppskk/images/Logo/LOGO_UMS_hitam.png

**KK14203 PENGATURCARAAN BERORIENTASIKAN OBJEK (2019-2020)**

**Seksyen: 1**

**Nama pensyarah: SAMRY@MOHD SHAMRIE SAININ**

**PROJECT 2**

|  |  |
| --- | --- |
| NAMA | NO.MATRIK |
| Mohd Hisyamudin bin Abd Aziz | BI19110028 |

1. **Java Code**

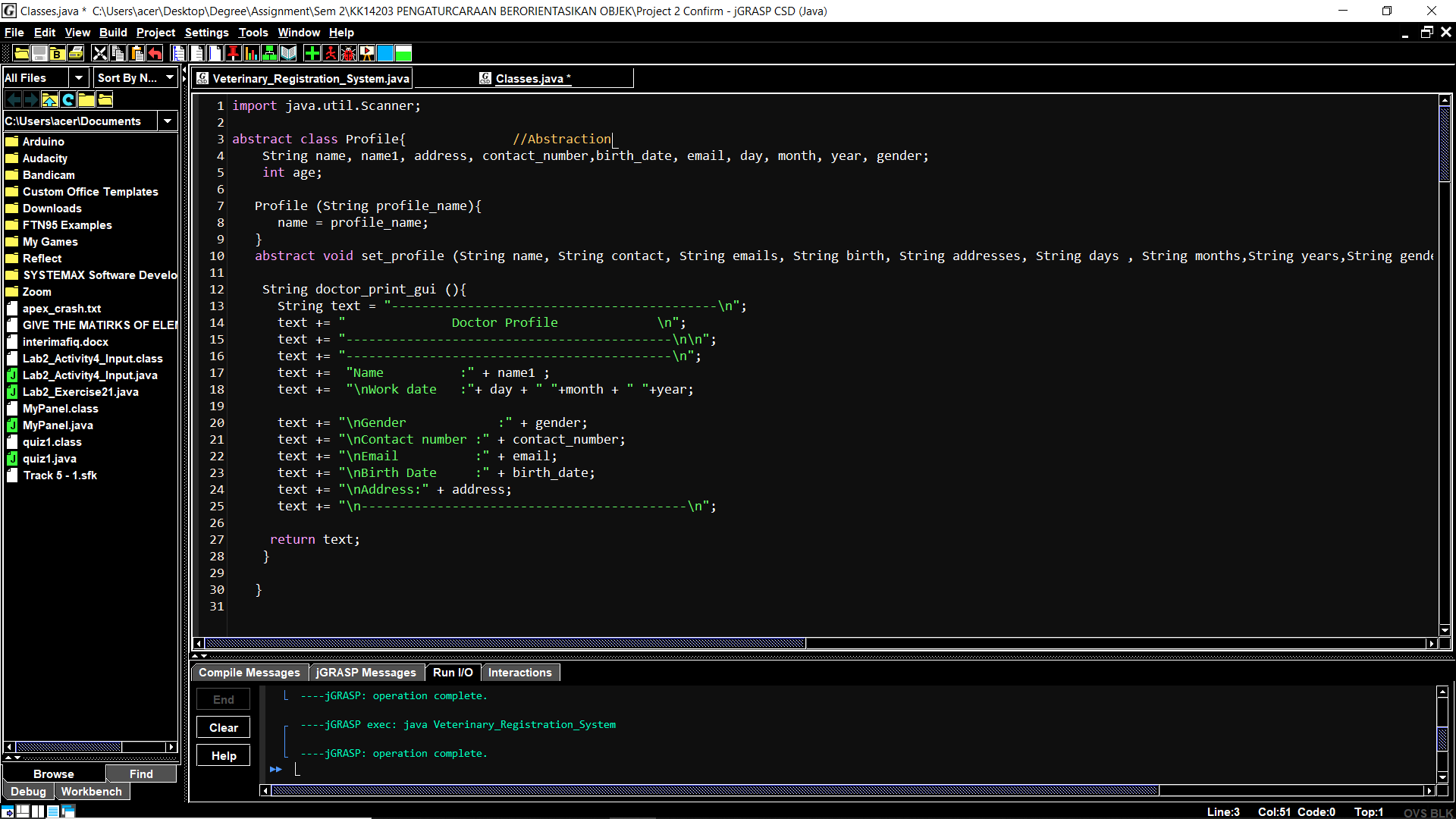
Link to access it in GitHub:

<https://github.com/Hisyamudin/KK14203-OOP/tree/master/Project%202>

1. **Object Oriented Concept Implementation**

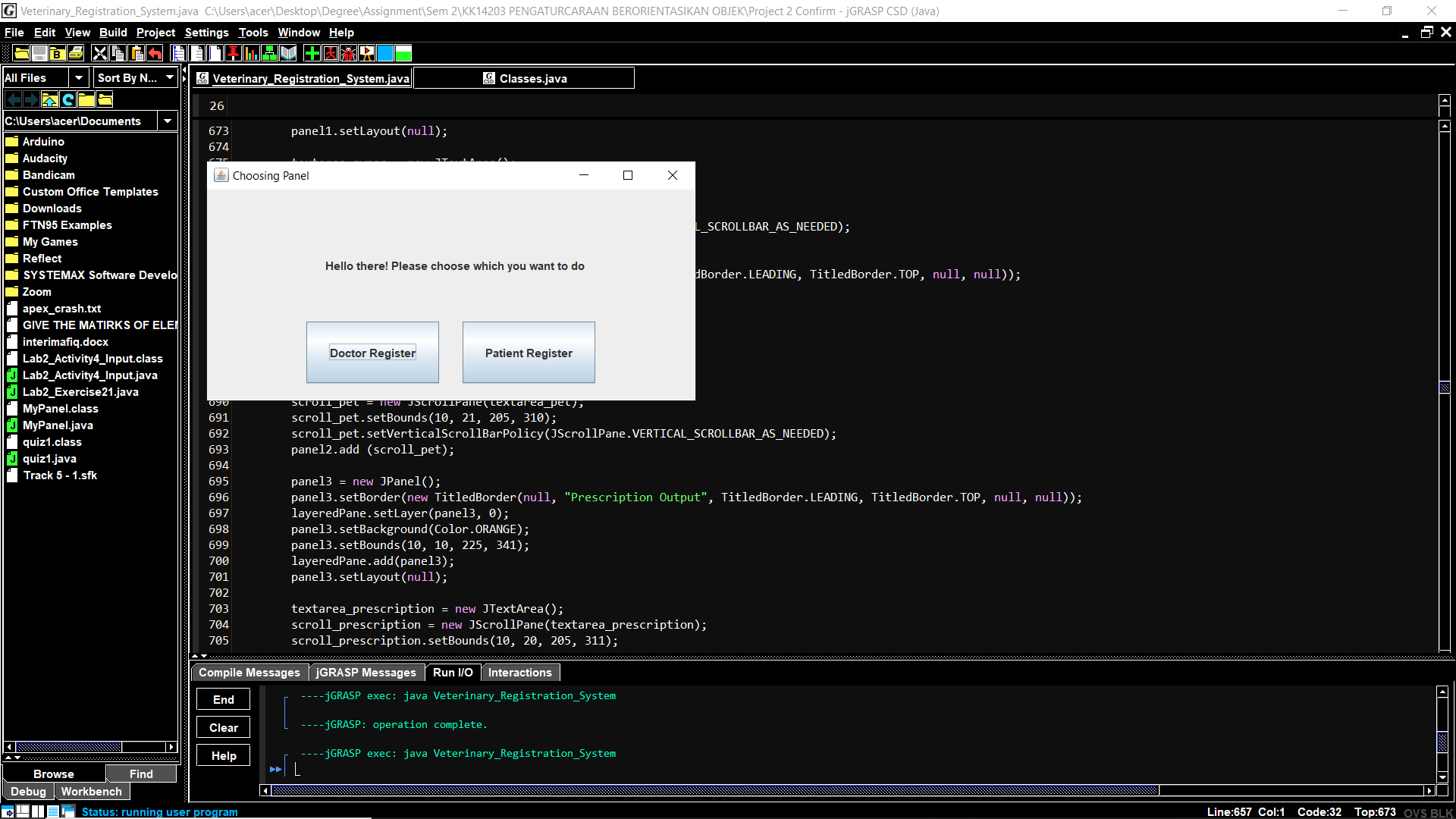
**2.1 Abstraction**

Abstraction aims to hide complexity from the users and show them only the relevant information. In this project, the abstraction is implemented to create a profile of a person like, Doctor’s profile. The abstract class is called Profile. So, if we want to develop this project, we can add the staff, cleaner and member’s profile into the project with ease.



**Figure 1**: Code of implementation of Abstract with abstract class Profile

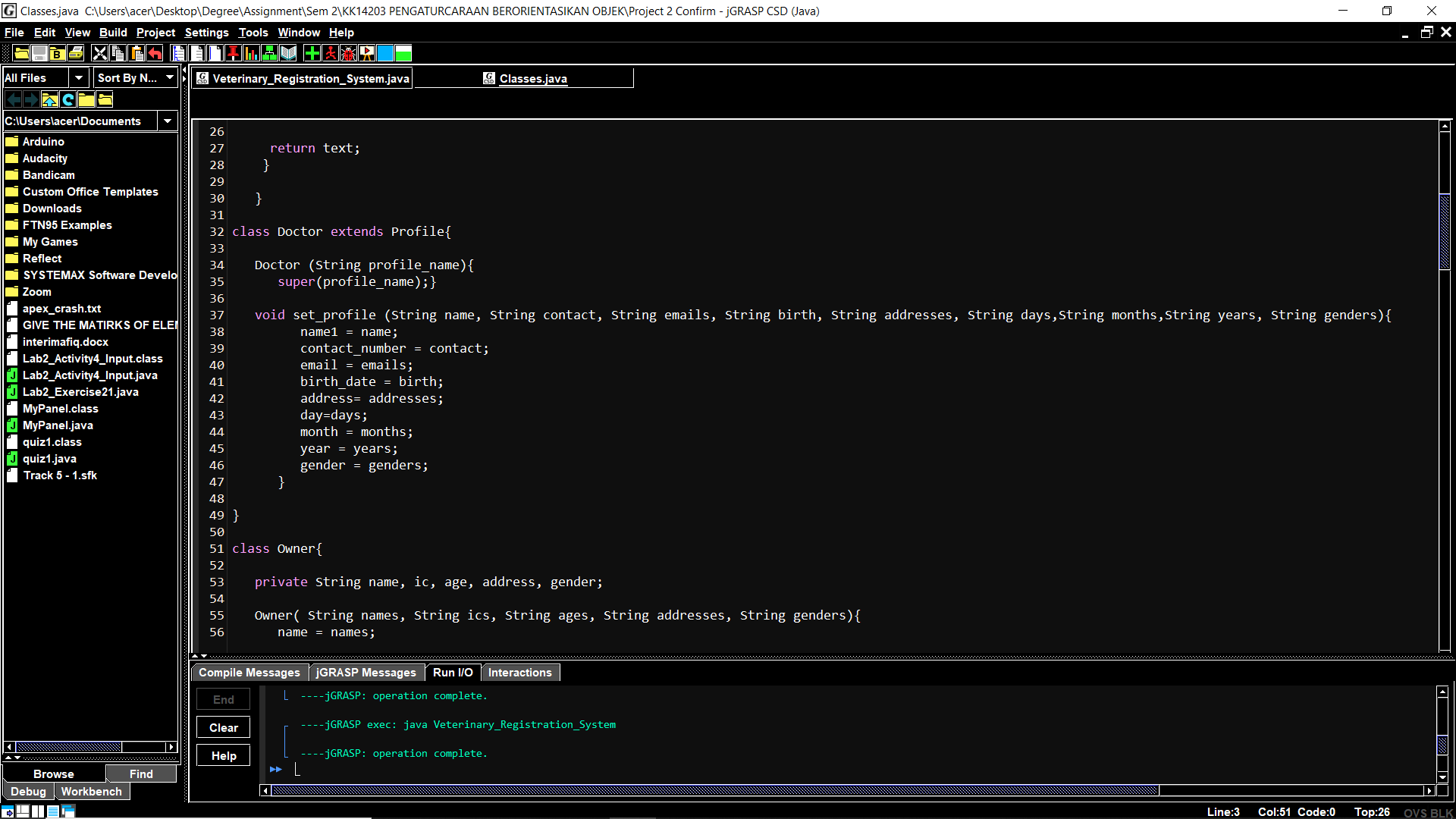
So, after the class is created, we can add another class that use this as their parent class to inherit it.



**Figure 2:** Example of the program to add the profile of a staff which is doctor in the clinic.

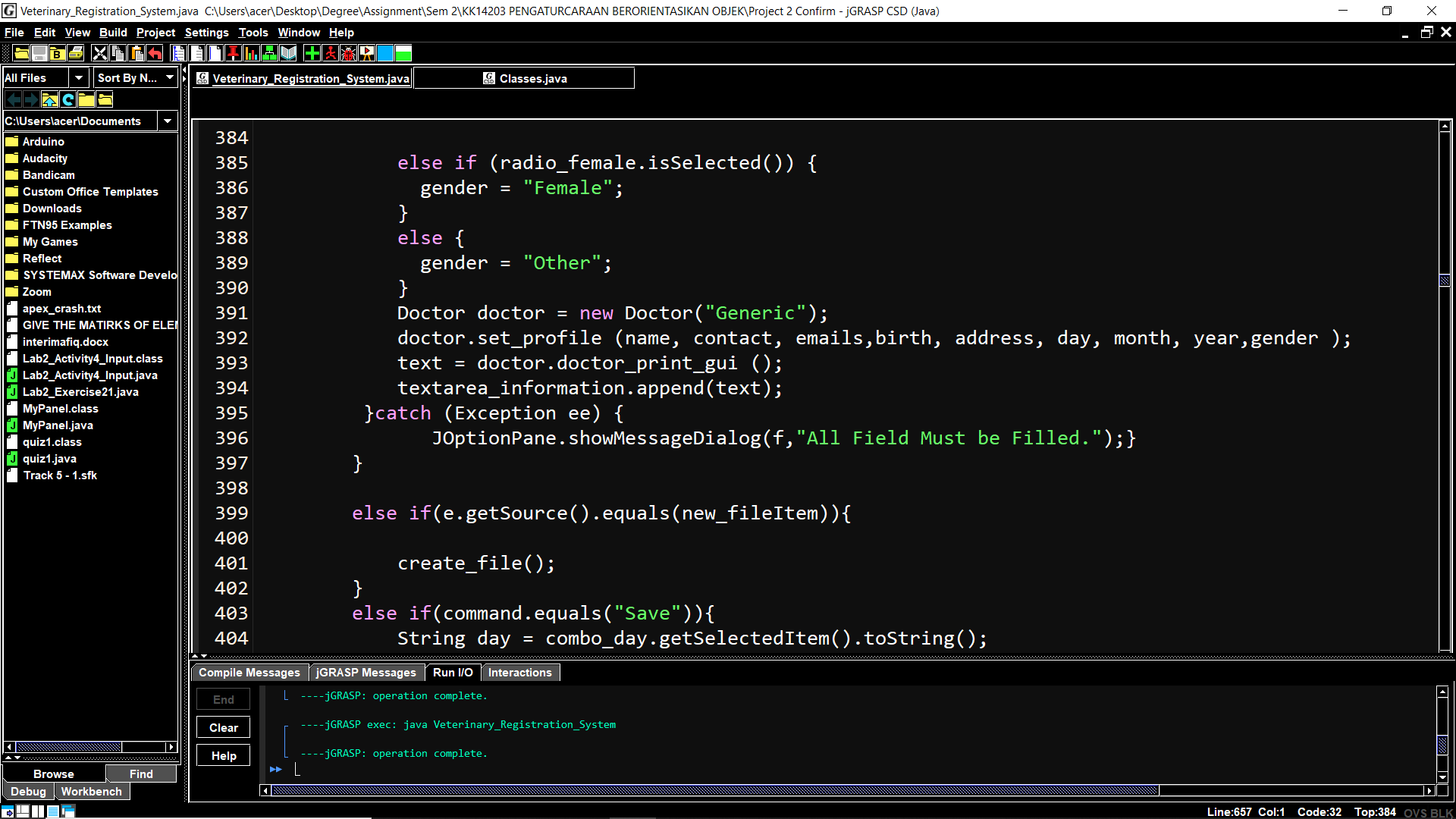
**2.2** [**Polymorphism**](https://www.geeksforgeeks.org/polymorphism-in-java/)

Polymorphism refers to the ability of Object Oriented Programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of these entities.



**Figure 3:** The class Doctor override the Profile class.

There is two type of polymorphism in object oriented programming which is method overloading and method overriding.The polymorphism that this program use is method over riding which is same method isoverriddenwith same signature indifferent classes. Type of object on which method is being invoked is not known at compile time but will be decided at run time. In this program, the Profile is a parent class and Doctor s a child class. The child class is overriding the method set\_profile() of parent class so that it can use for it self.

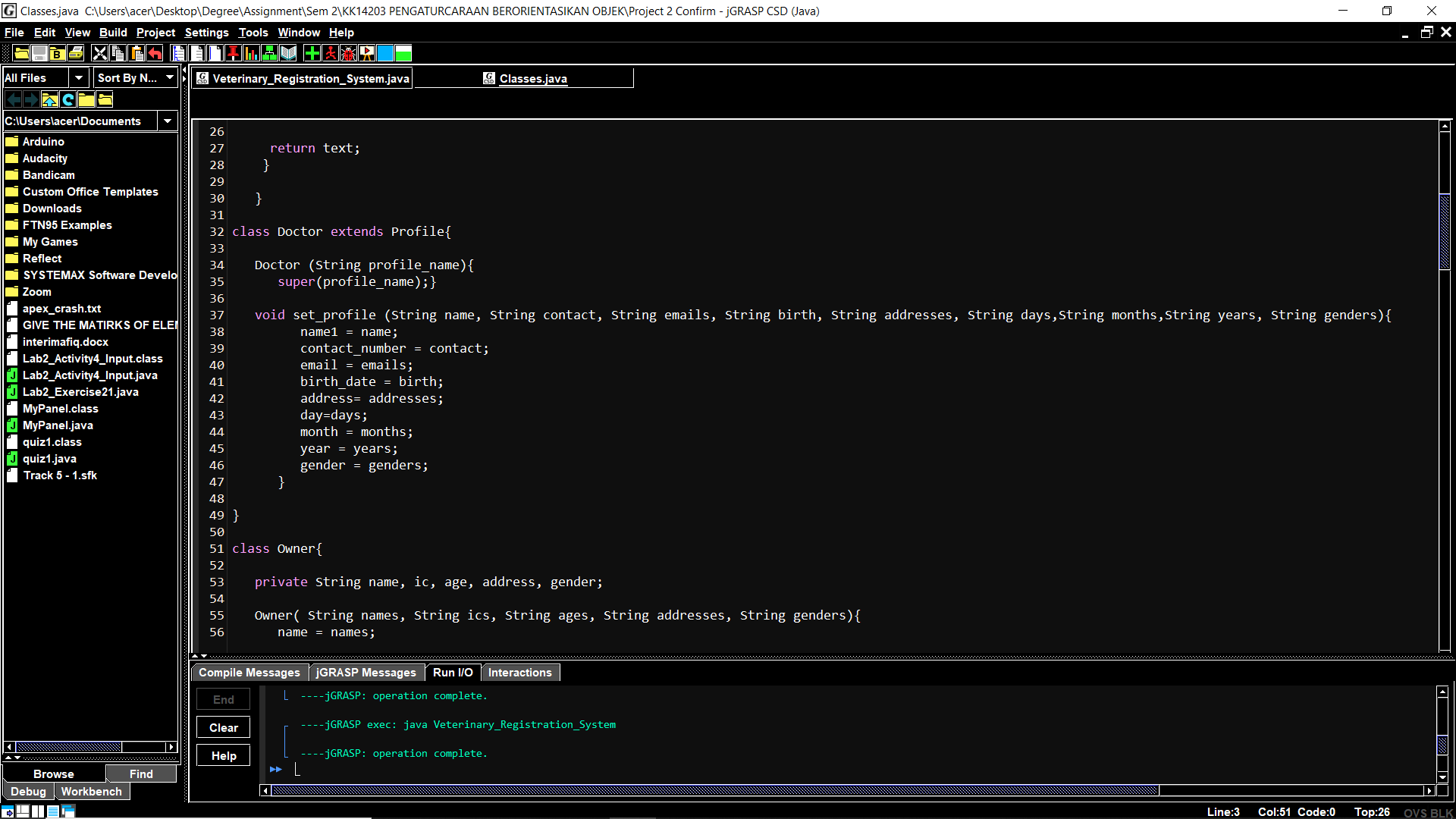


**Figure 4:** The set\_profile method in Doctor class would be called in line 392.

In figure 4, it shows that the set\_profile method in Doctor class would be called instead of set\_profile method in Profile class. This means that, it can ease us for the coder who need to use many method with same name to do a program and different methods of the same name can be called from the object.

**2.3 Inheritance.**

Inheritance is to create a child class that inherits the fields and methods of the parent class. The child class can override the values and methods of the parent class. It also can add new data and functionality to its parent. First and foremost, A class which is child class can extend another class parent class by inheriting its features.

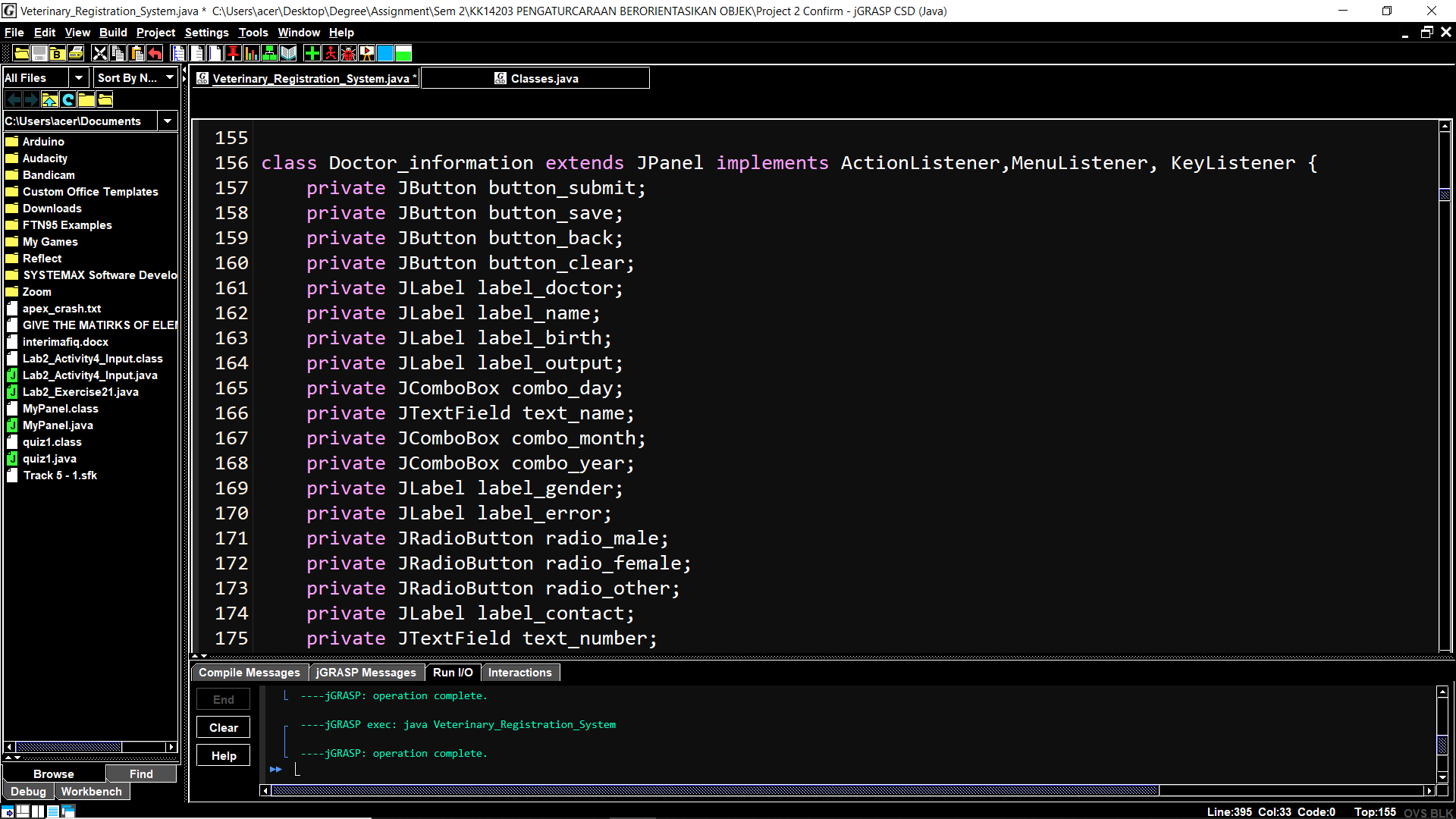


**Figure 5:** class Doctor inherit features from Profile Class.

In figure 5, it shows that, class doctor will be constructed based on Profile who will become a parent class. This means that, the Doctor class will inherit features from Profile class and can use some of the features there. After that, Inheritance supports the concept of “reusability”. For an example, when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

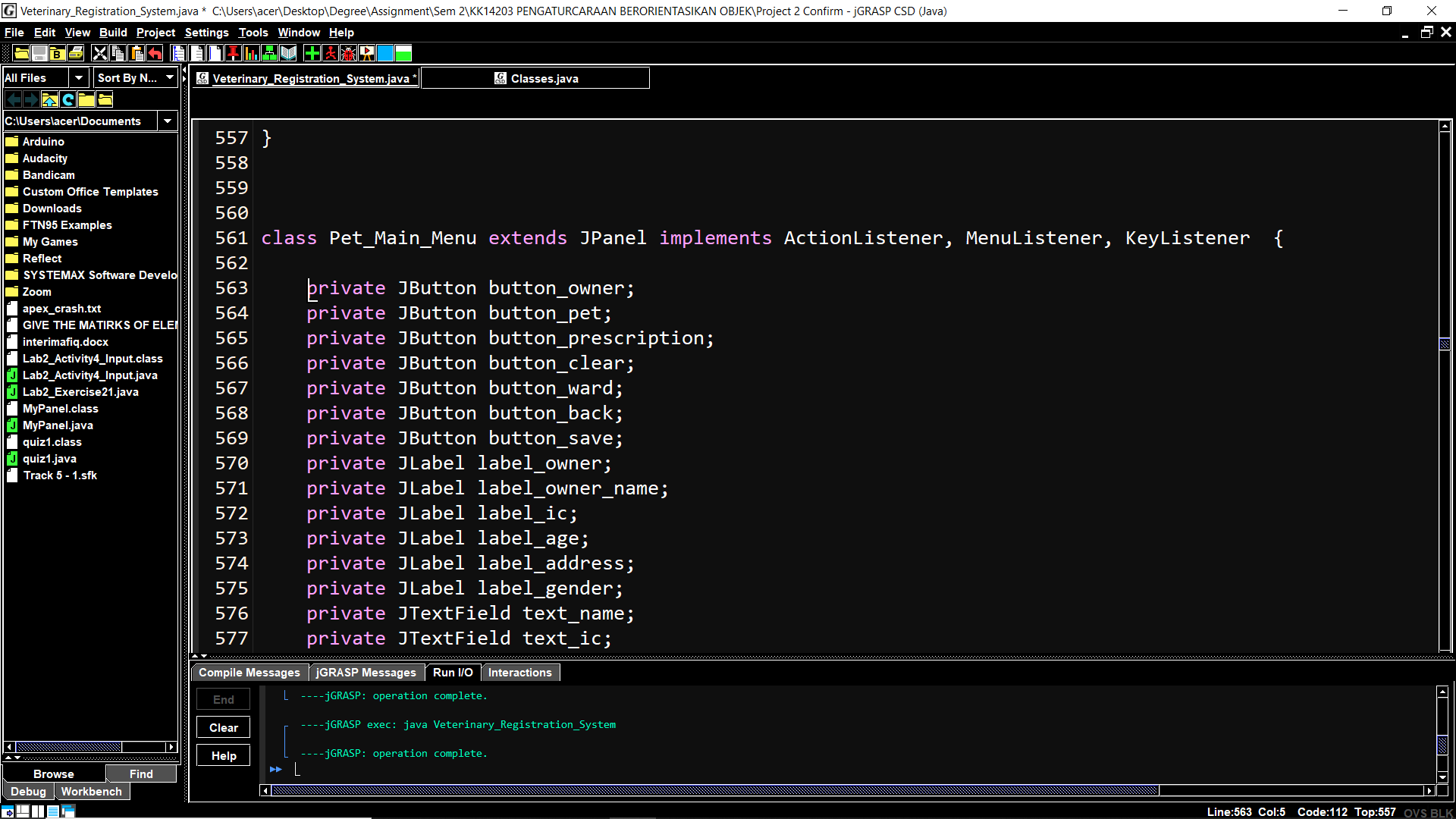
**2.4 Encapsulation.**

Encapsulation is to hide the implementation details from users. If a data member is private, it means that it can only be accseed within the same class. No outside class can access private variable of other class. It’s a protective barrier that keeps the data and code safe within the class itself.



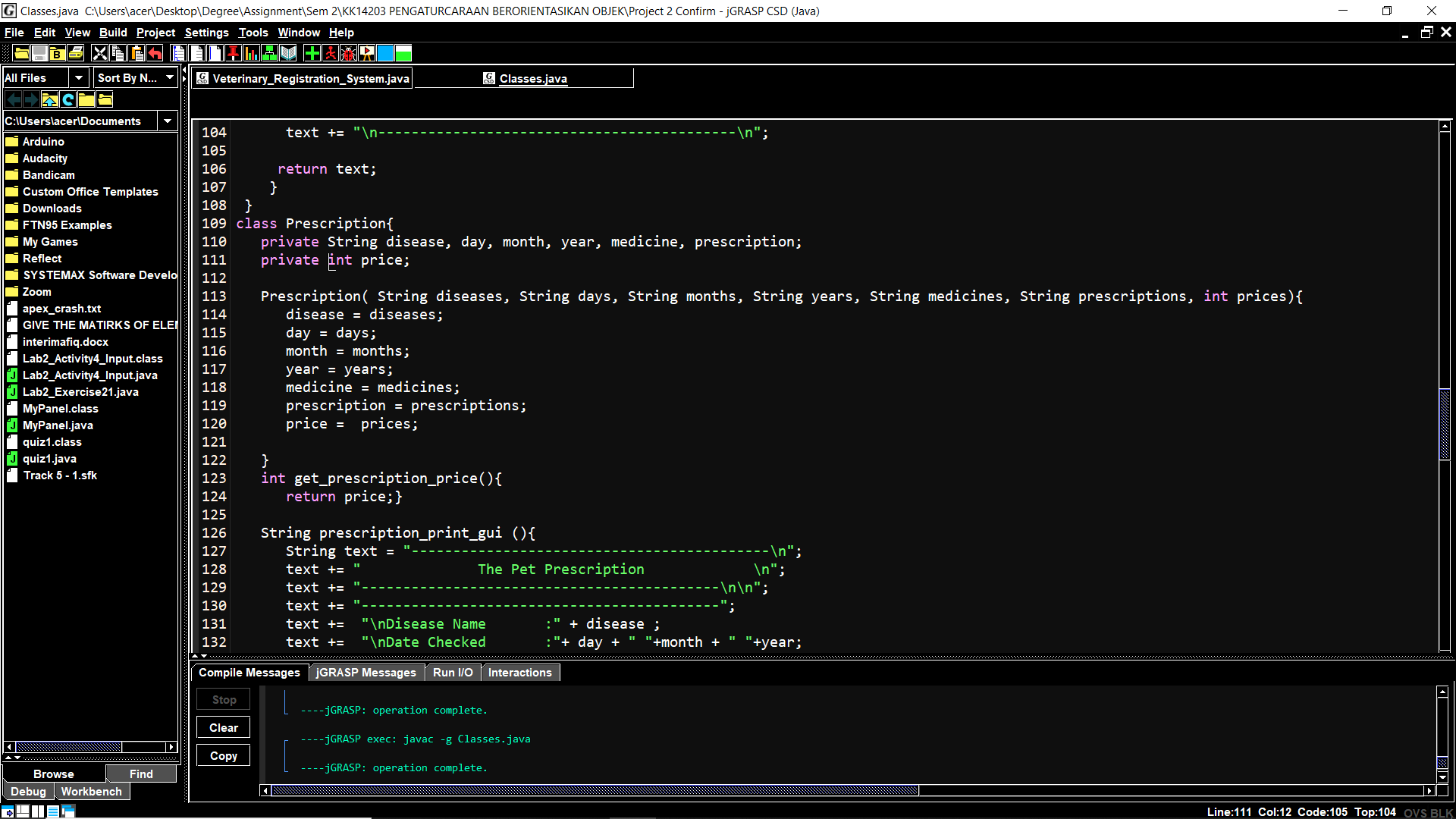
**Figure 6:** Private variable in Doctor\_information class.

Figure 6 shows that, the private variable means that the variable only can be accessed in the Doctor\_information class and can not be accessed outside of the class. The advantages of using this encapsulation is, It improves maintainability and flexibility and re-usability. For an example, the same varible name can be use in same program without having any problem.



**Figure 7:** Same variable used in other class.

Figure 7 shows that, the same variable name can be used in other class like the varible button\_save. The button\_save variable were bothused in my the Doctor\_information class and Pet\_Main\_Menu class and can be ran without any problem. After that, the advantages is the User would not be knowing what is going on behind the scene. They would only be knowing that to update a field call set method and to read a field call get method but what these set and get methods are doing is purely hidden from them. So if we want to get the variable value, we can use write only method by return the variable only.

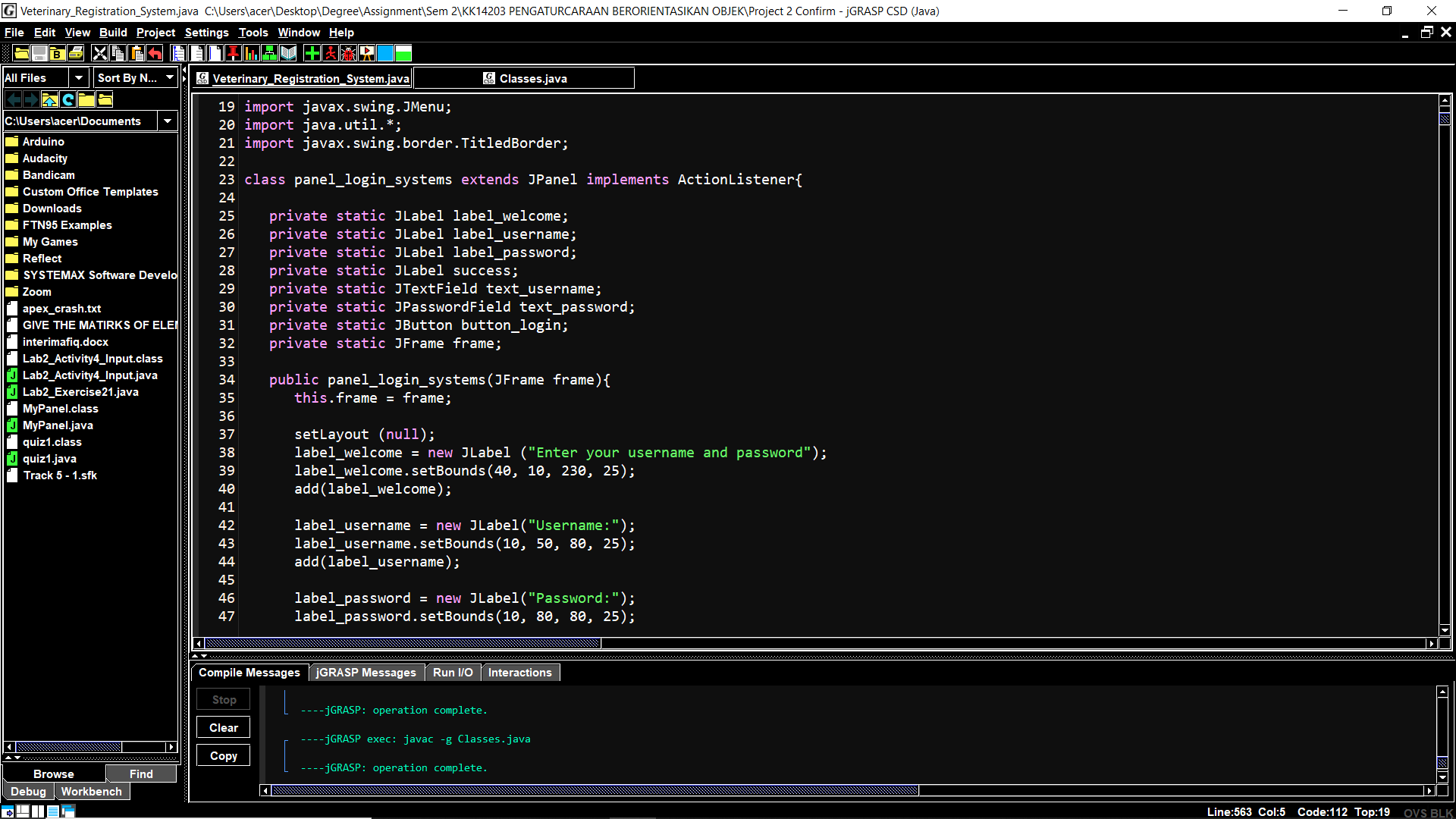


**Figure 8:** Get the variable price value

In my program, the variable price is private so to get the value, the get\_prescription\_price() method is created to return the value. So, if the price is needed to be used, we can call the method to get the variable value.

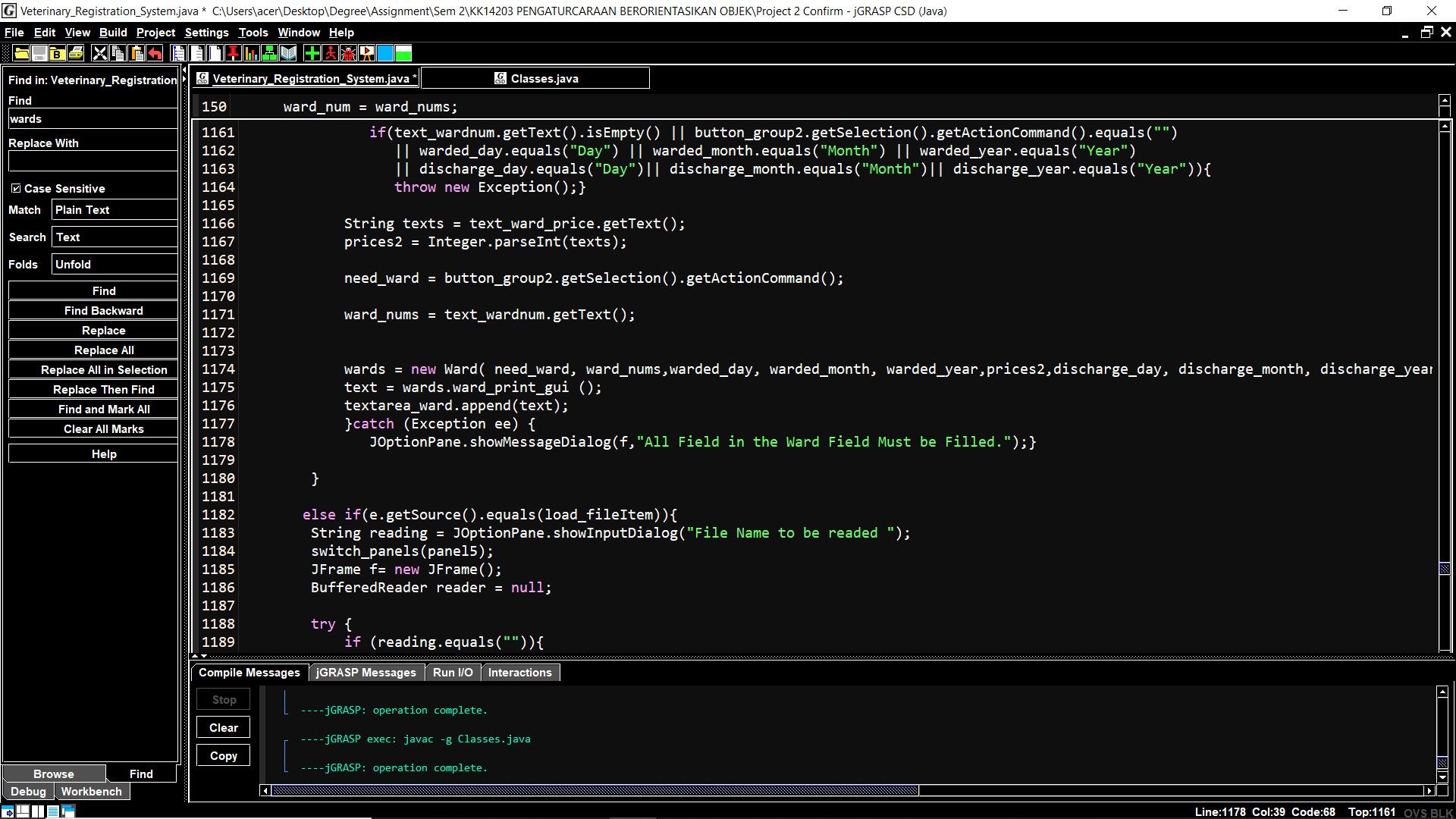
**2.5 Object & Classes**

**Class**are a blueprint or a set of instructions to build a specific type of object. It is a basic concept of Object-Oriented Programming which revolve around the real-life entities. Class in Java determines how an object will behave and what the object will contain. Meanwhile **Object** is an instance of a class. An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. In my project, there were total of 10 class used which is, Profile, Doctor, Owner, Pet, Prescription, Ward, Pet\_Main\_Menu,Doctor\_information, choosing\_panel and panel\_login system\_class.



**Figure 9:** panel\_login\_system class.

Figure 9 shows that, the panel\_login\_system class is created at line 23 which mean that it contain some method can be used later if object is created for it. After that, there were total of 5 object that is created which is choose\_panel, doctor\_panel, pet\_panel, doctor, owners, pets, prescriptions and wards.

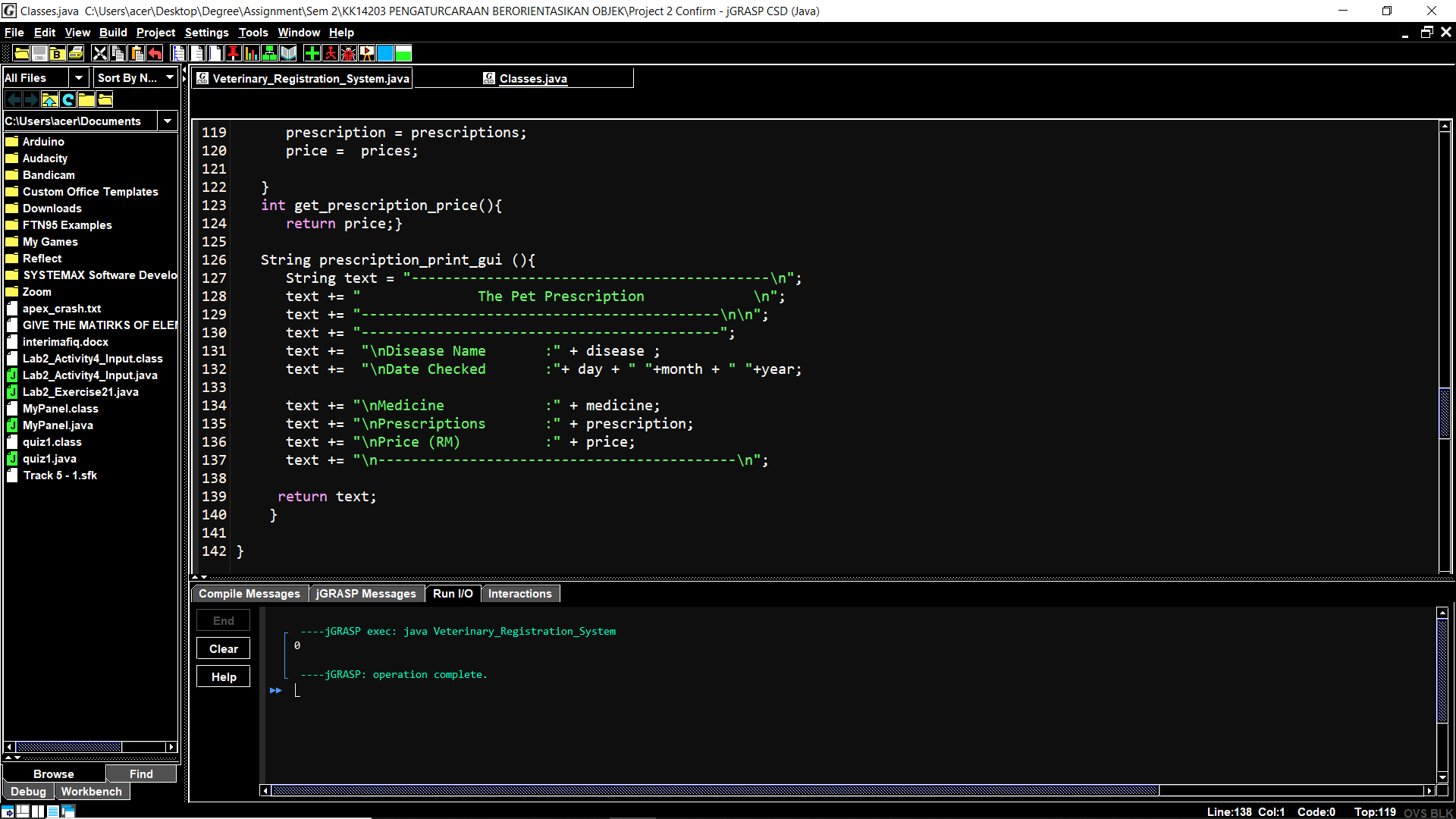


**Figure 10:** wards object created.

In figure 10, the wards object is created which means that it can be used to access the method in the Ward class. Classes are very useful in programming because we just can use it again to make a new object like if we need to create profile for 100 person, we can use it the class again and again by just creating a new object for it.

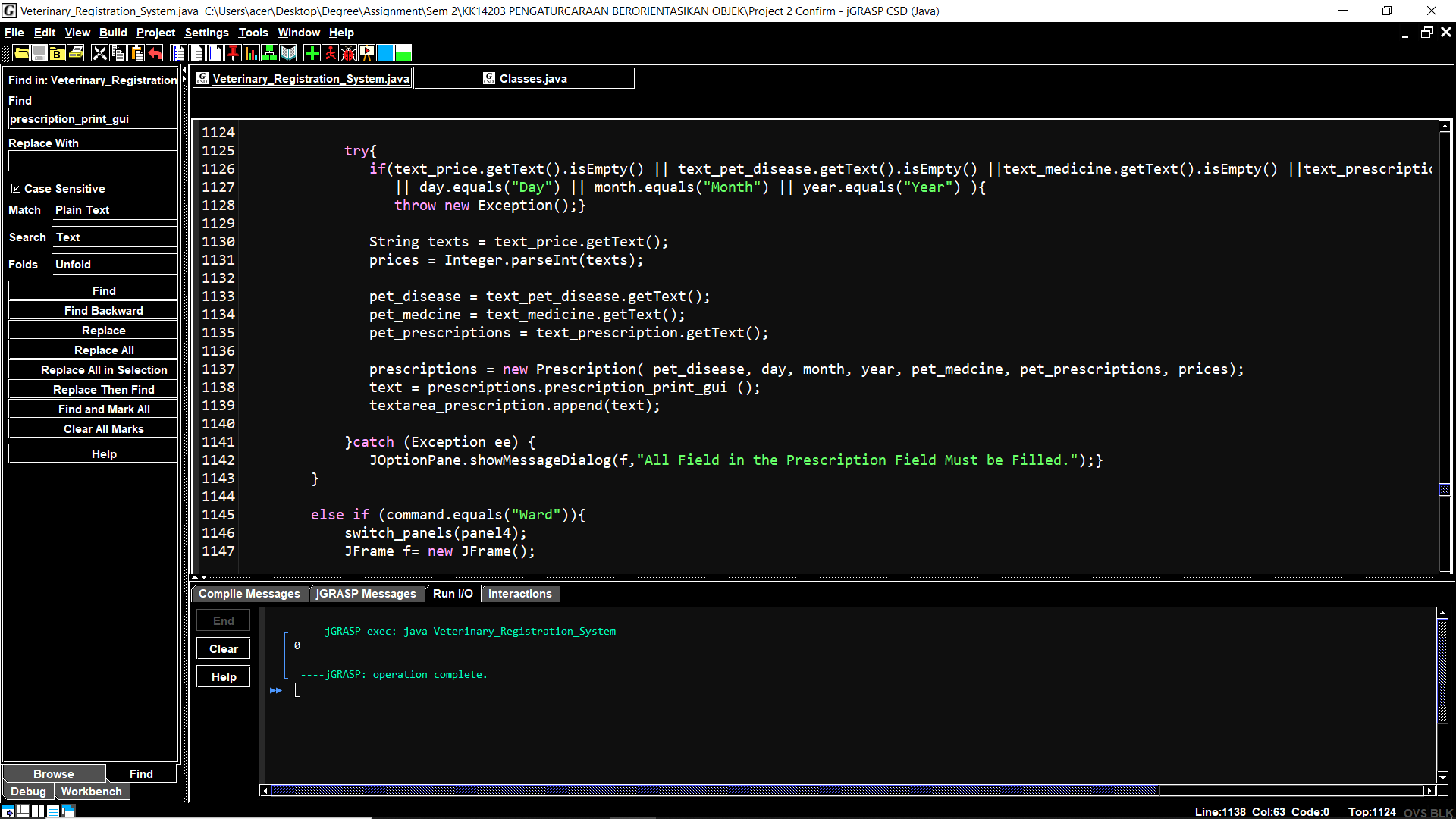
**2.6 Method.**

 A method is a collection of statements that perform some specific task and return result to the caller. A method can perform some specific task without returning anything. Methods allow us to **reuse** the code without retyping the code. In the program, there are some method that can return the result to the caller like the prescription\_print\_gui which return string value.



**Figure 11:** prescription\_print\_gui method return a String value.

In figure 11, a method called prescription\_print\_gui method in line 126 will return a String value which will print all the component in the variable string text. If the method is called, the output will be a string that the user have input into the variable.



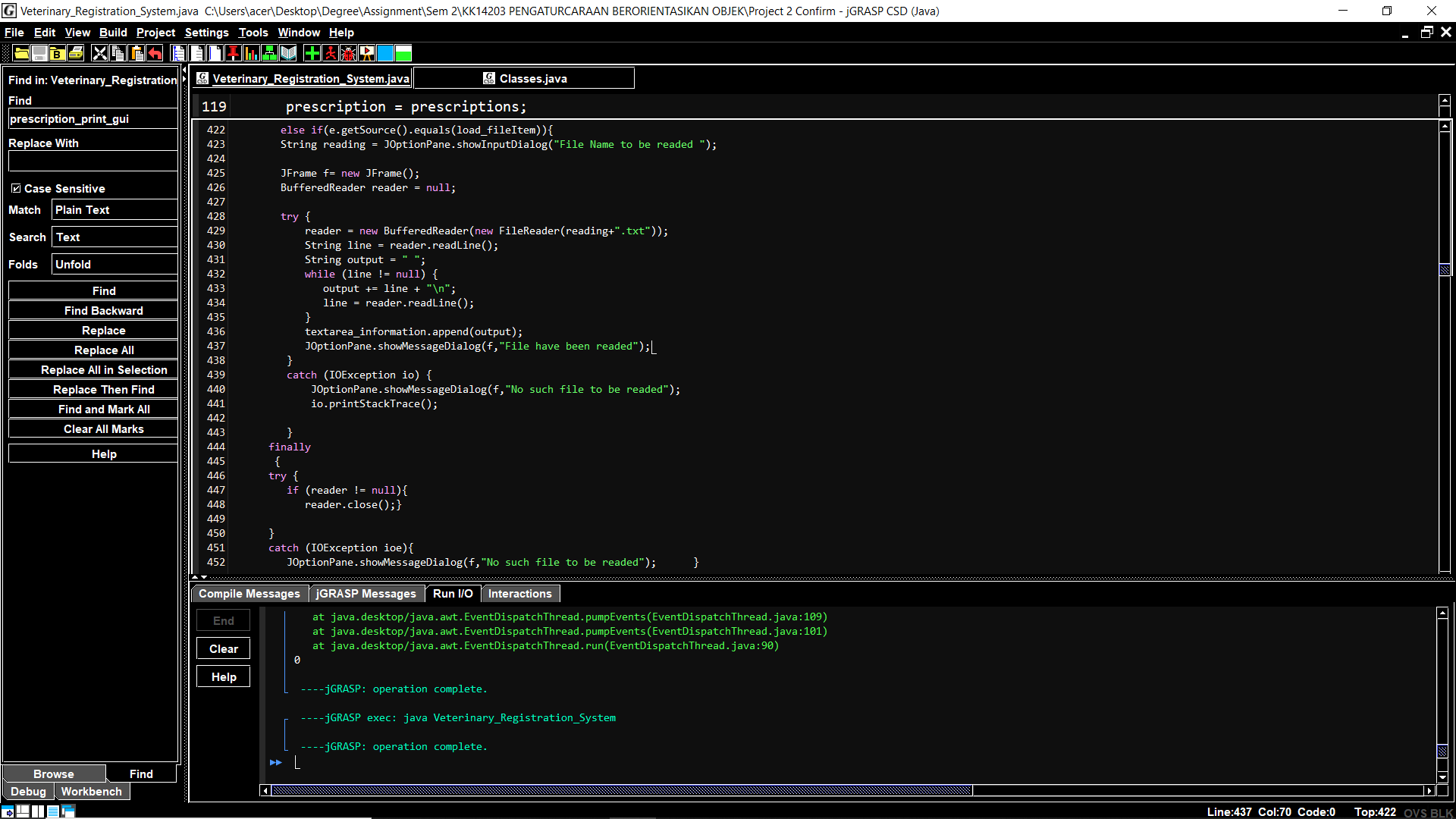
**Figure 12:** prescription\_print\_gui is called with prescriptions object.

In figure 12, the method prescription\_print\_gui is called in line 1138 with prescriptions object to print an output using the method. By using method, the code is more easier to read and can make a code more shorter.

**3.0 Read and Write Implementation**

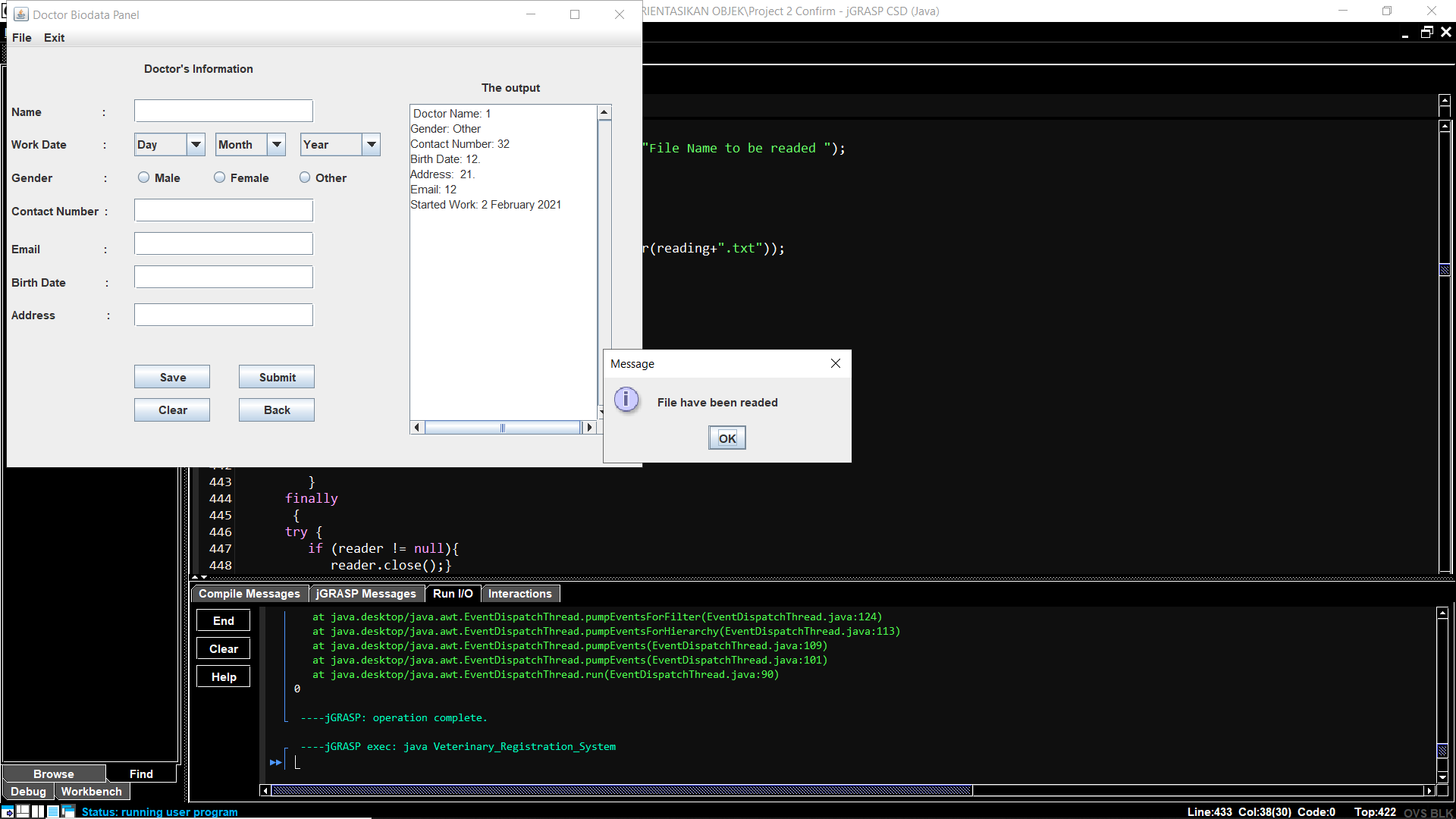
**3.1 Read Implementation.**

For a read implementation, it will read the input from external storage file which is text file. First and foremost, in the project, it will use an option pane which is to get an input string from the user. After that, in the program, there will be a bufferedreader and new file reader to read the file that the user have put the name before this.



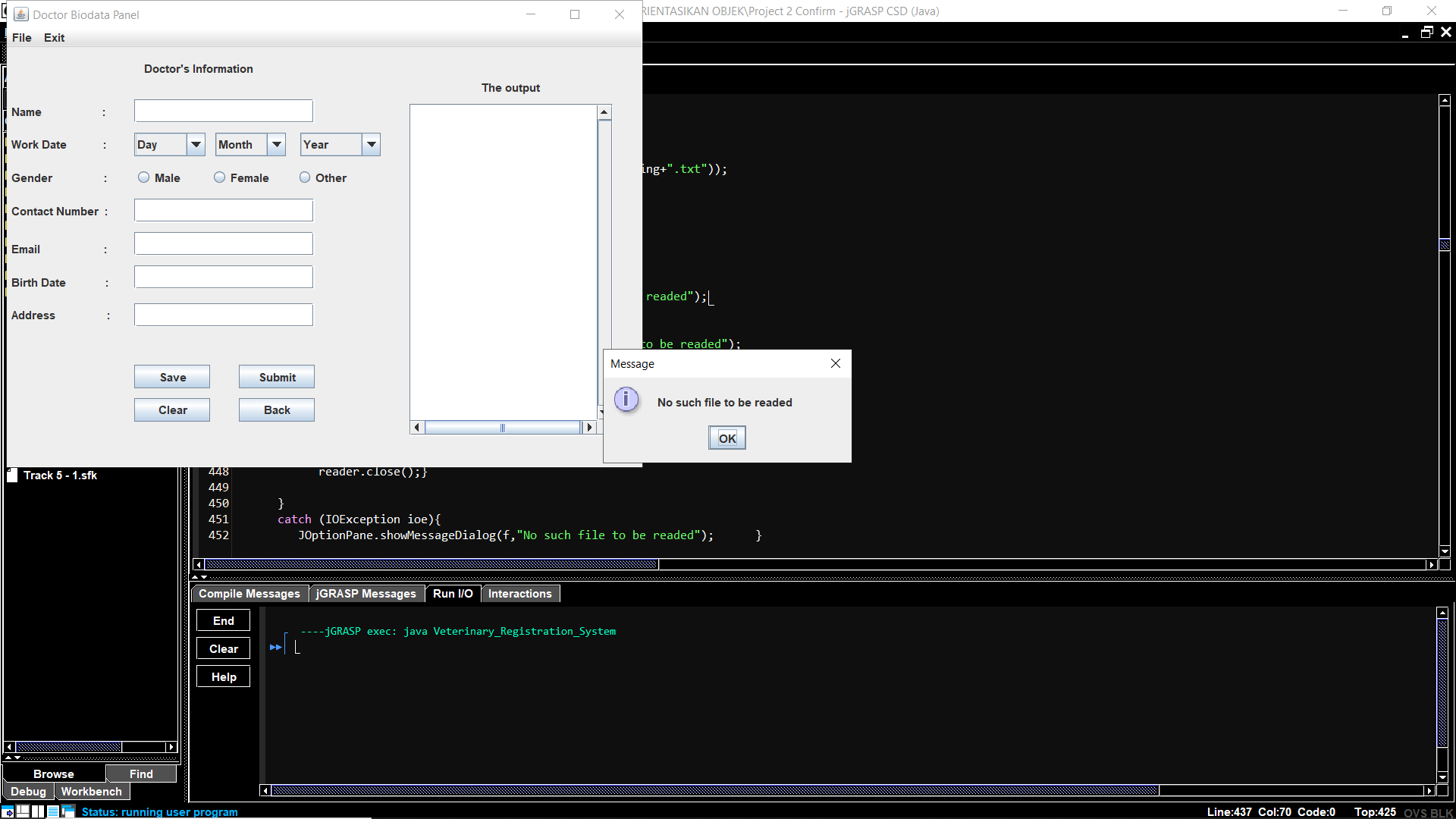
**Figure 13:** Code for read implementation.

In figure 13, it shows the full code for read implementation from line 422 to line 453. If the input from user have a name that match a file, the input will be read from the file.



**Figure 14:** Example if the file were readed.

In the same time, if the file is readed, a message will appear saying “file have been readed”.

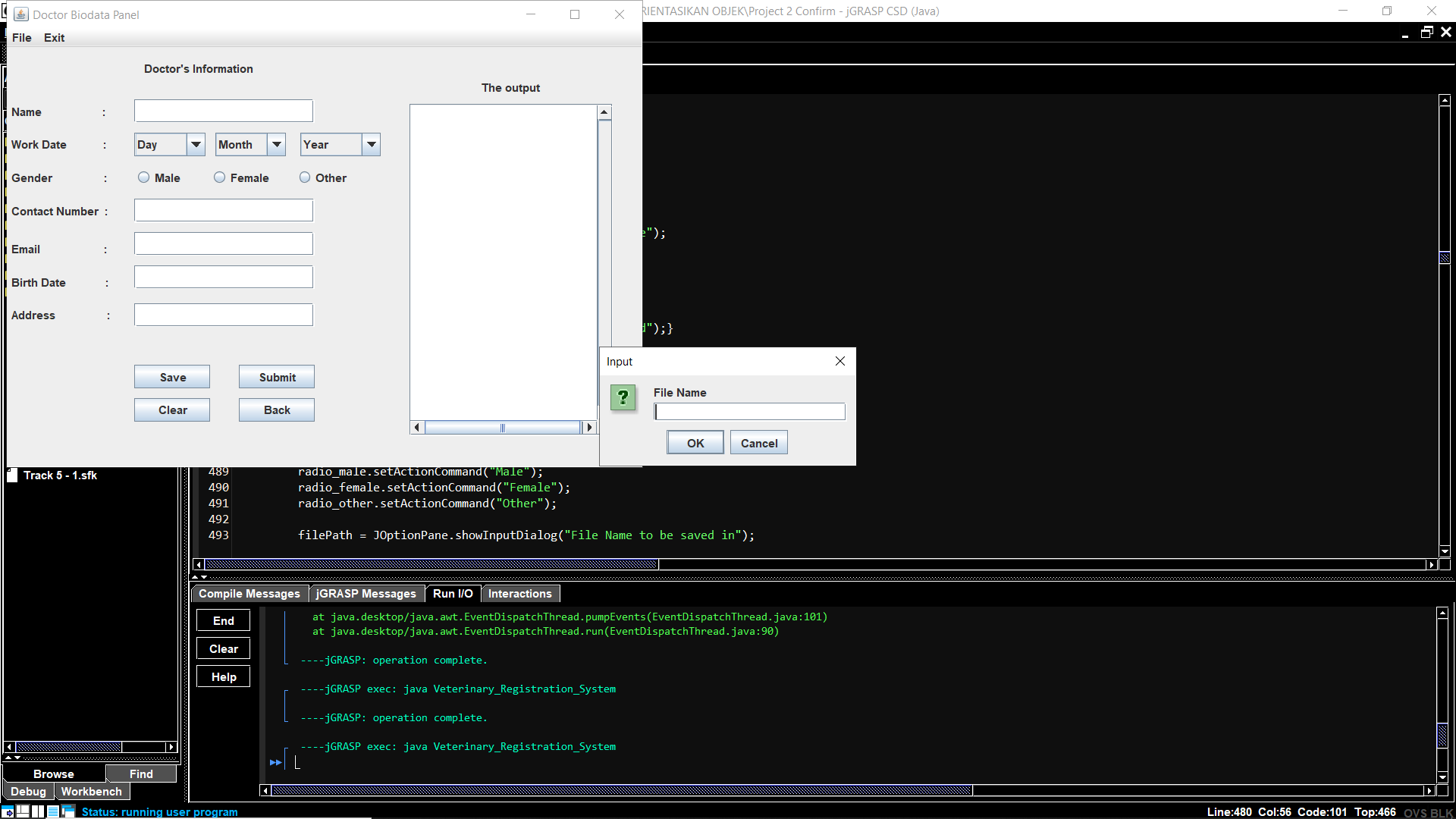


**Figure 15:** Example if there is no file were detected.

If there is no file to be readed, a message will appear saying “No such file to be readed”.

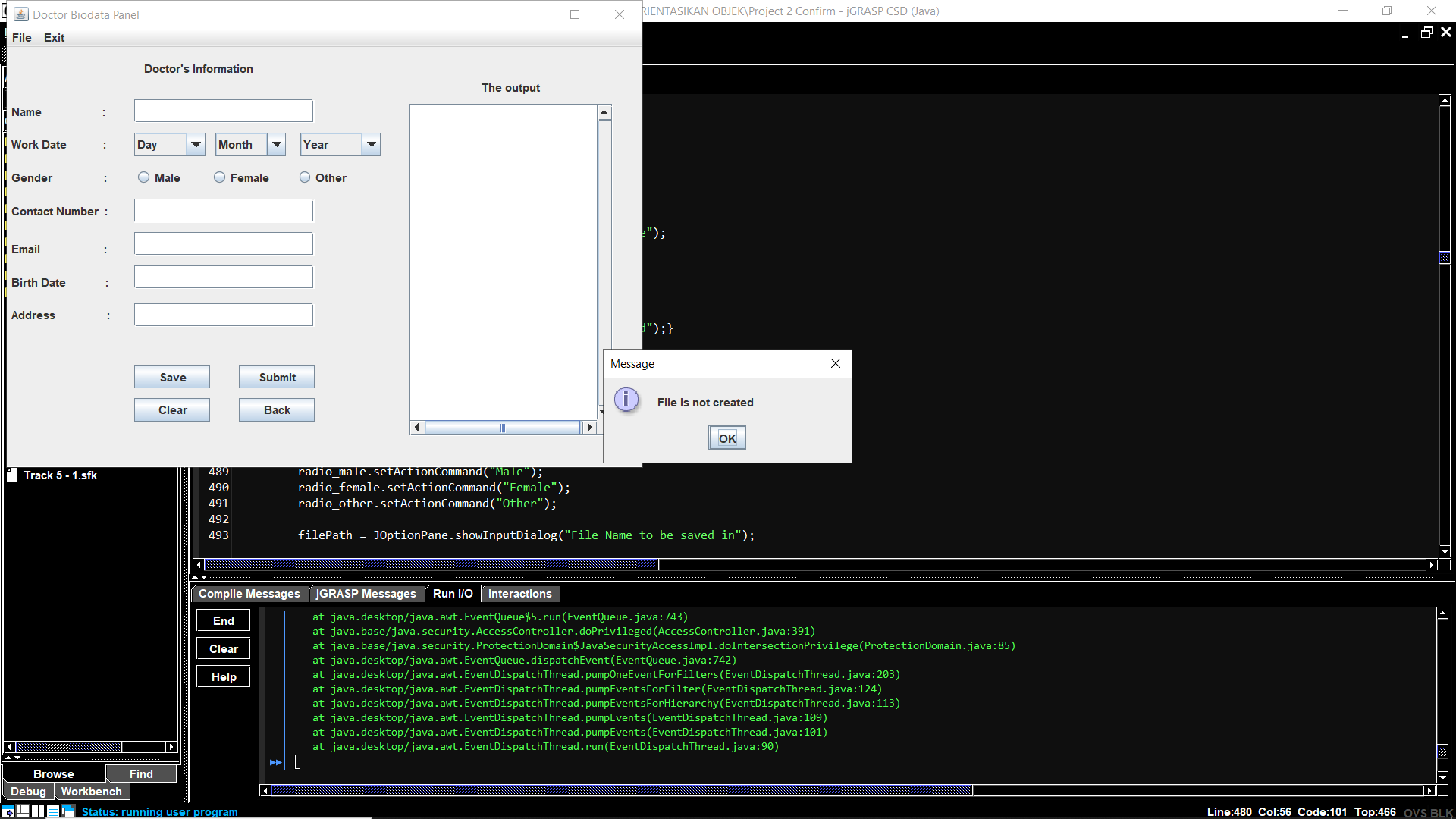
**3.2 Write Implementation.**

For a write implementation, it require the program to write the output to external file. So in the program, to write into a file, a file must be created first. In the project, to create a file, we can either choose from the menu to create a new file or just click the save button and it will create a file automatically. If a user click from the menu to create a new file, there will be an message dialouge to let us input the name file. Besides, the file is created by using a formatter.



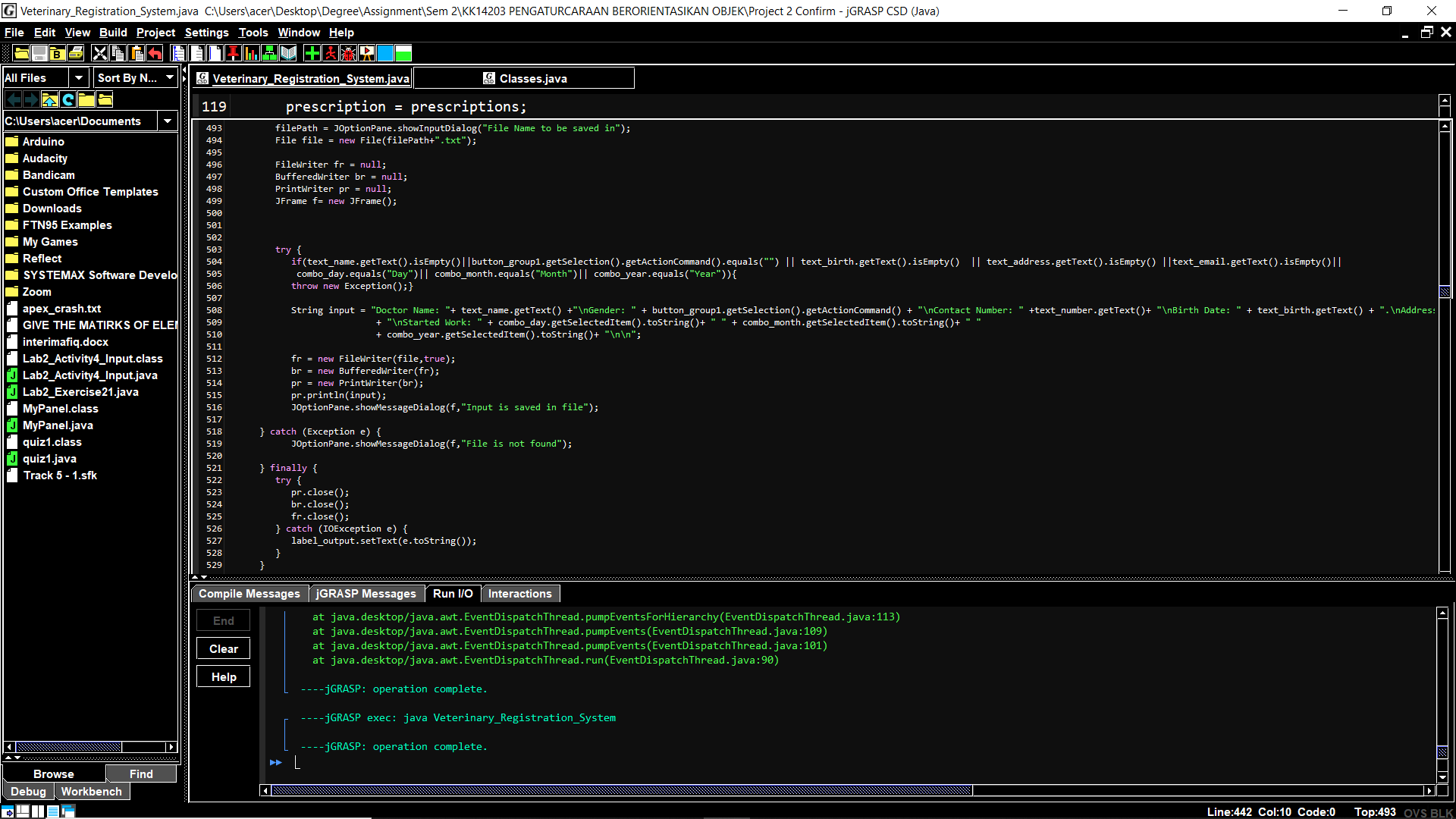
**Figure 16:** An Input dialogue will appear to let the user enter an input.

If the user let the field empty or click cancel, a message dialogue will appear and says”File is not created”.



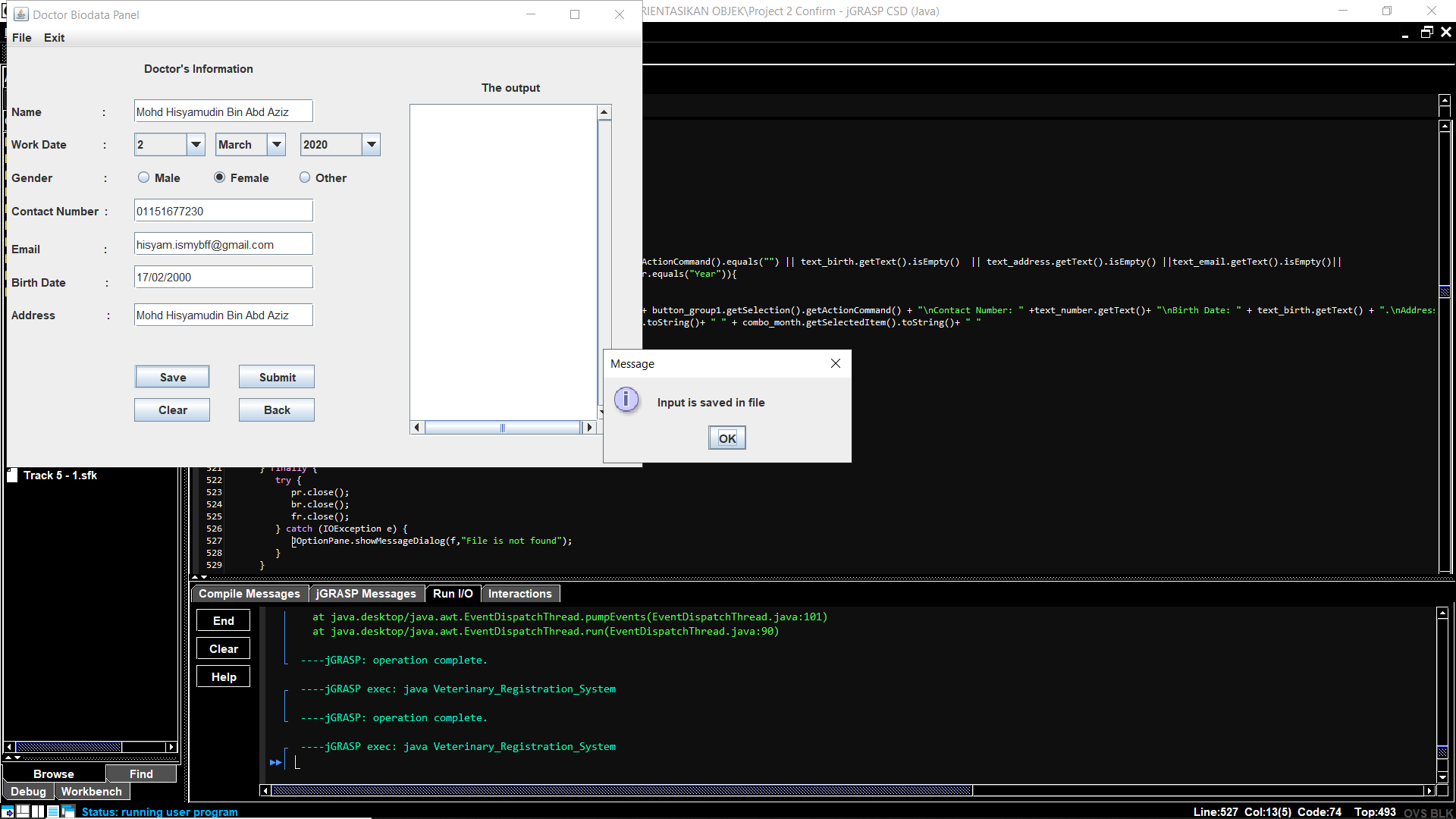
**Figure 17:** A message dialogue will appear if the user cancel or fail to put input.

To write the input from user into a file, the user must click the save button. After the user clicked the button, an input dialogue will appear to let the user user enter the file name to write the input in.



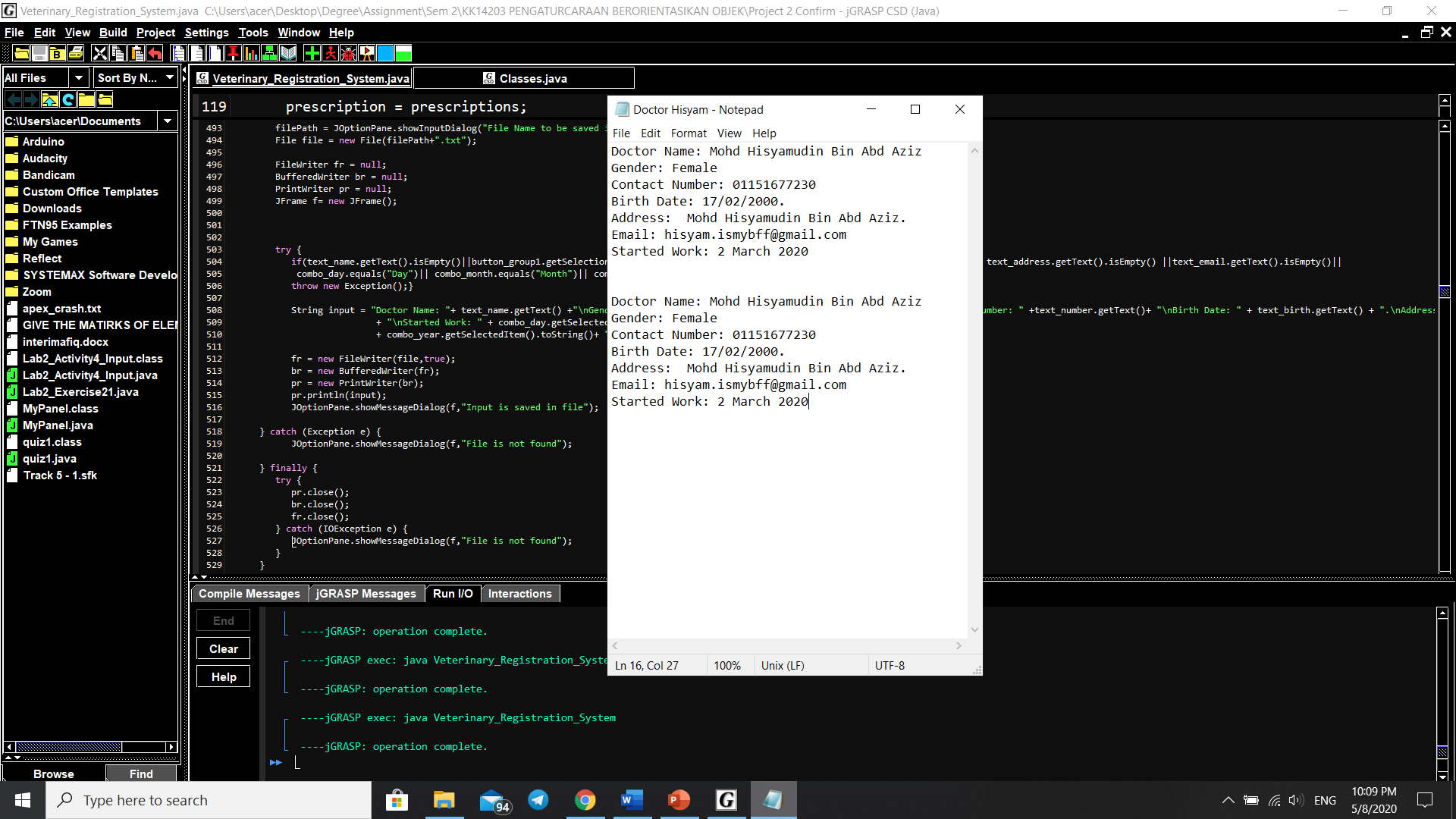
**Figure 18:** The full code of write implementation for Doctor.

Figure 18 shows the full code of write implementation which is from line 491 to line 529. First, a string which is the filePath variable will be an input from the user. Then, a new file will be created based on the input from the user. After that, a new Filewrite is needed to be created so it can write the input to the file. In the program the Filewriter is named fr and the fr will print all of the input to a file. Then, a string input is created so that the Printwriter can get the item from the string. Then, all of them needed to be closed so the operation ended and written on the file.



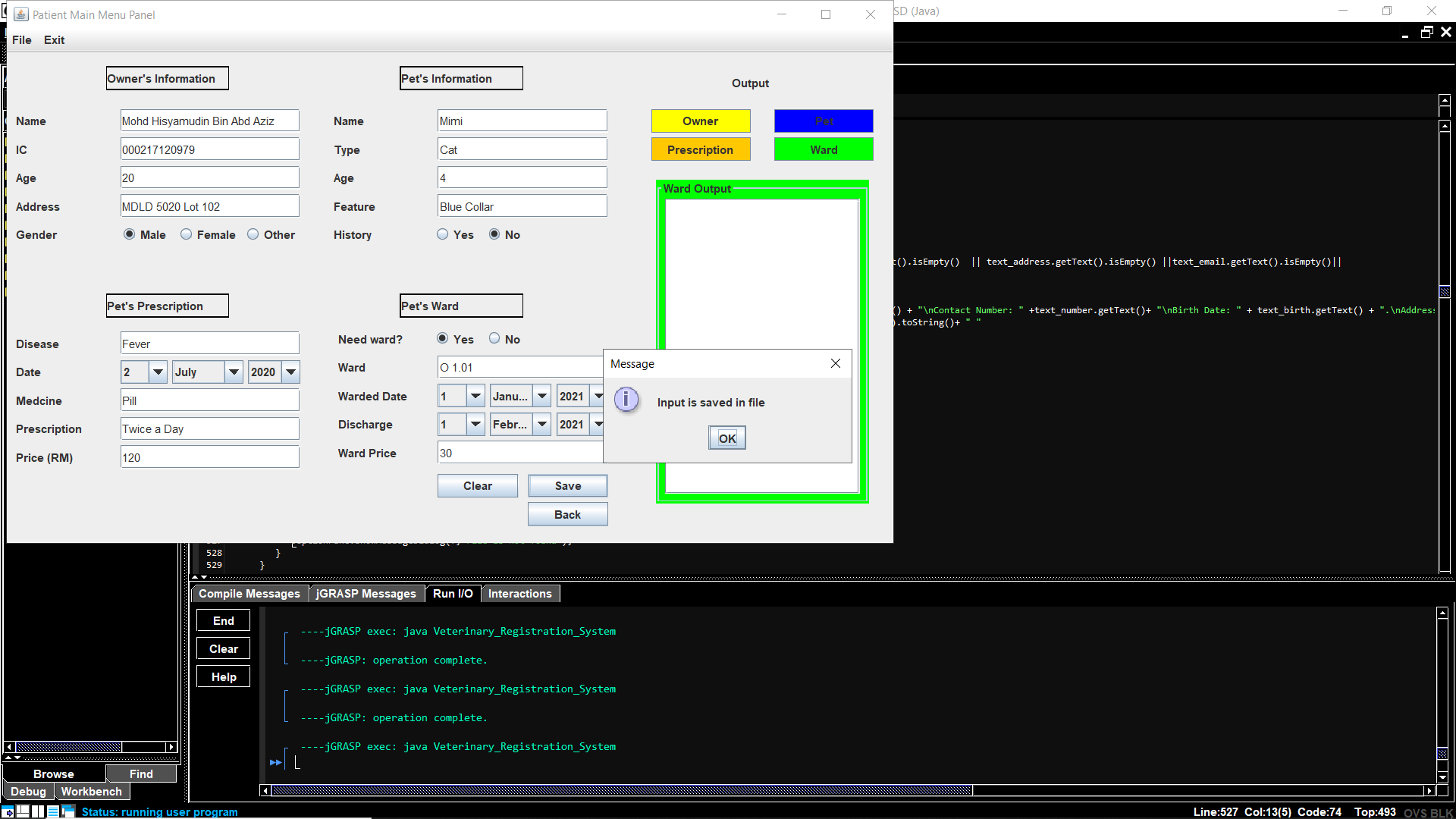
**Figure 19:** A message pane will appear if the input is writed.

In figure 18, if the input is writed successfully, it shows a message pane which is telling “Input is saved in file”.



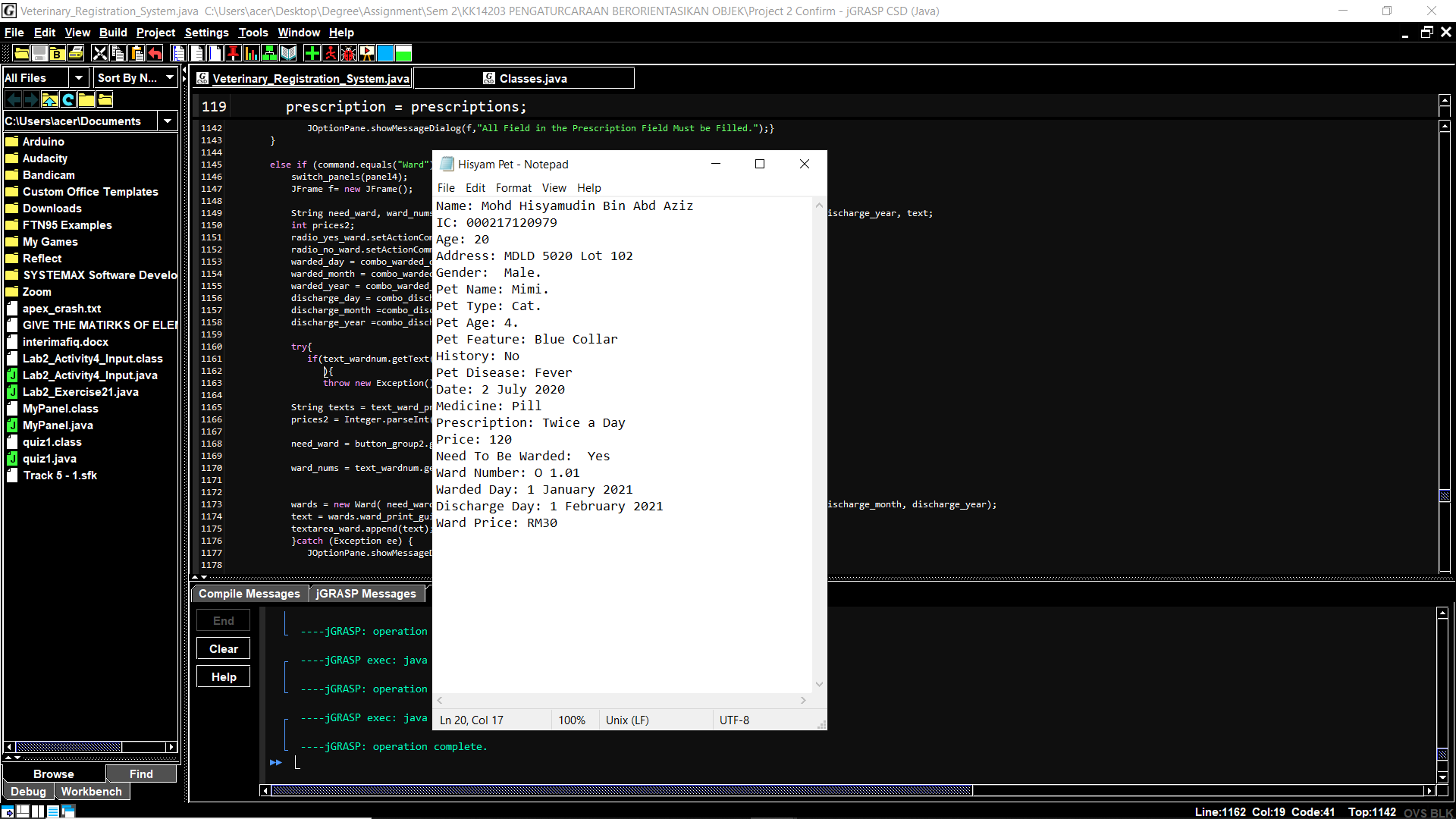
**Figure 20:** Example of how the input will be saved in the text file.

In figure 19, it shows that the information from the figure 18 is successfully writed in a text file named “Doctor Hisyam”.



**Figure 21:** A message pane will appear if the input is writed in a file.

Figure 21 shows, a patient file is succesfully saved into a file.



**Figure 22:** Example of how the input will be saved in the text file.

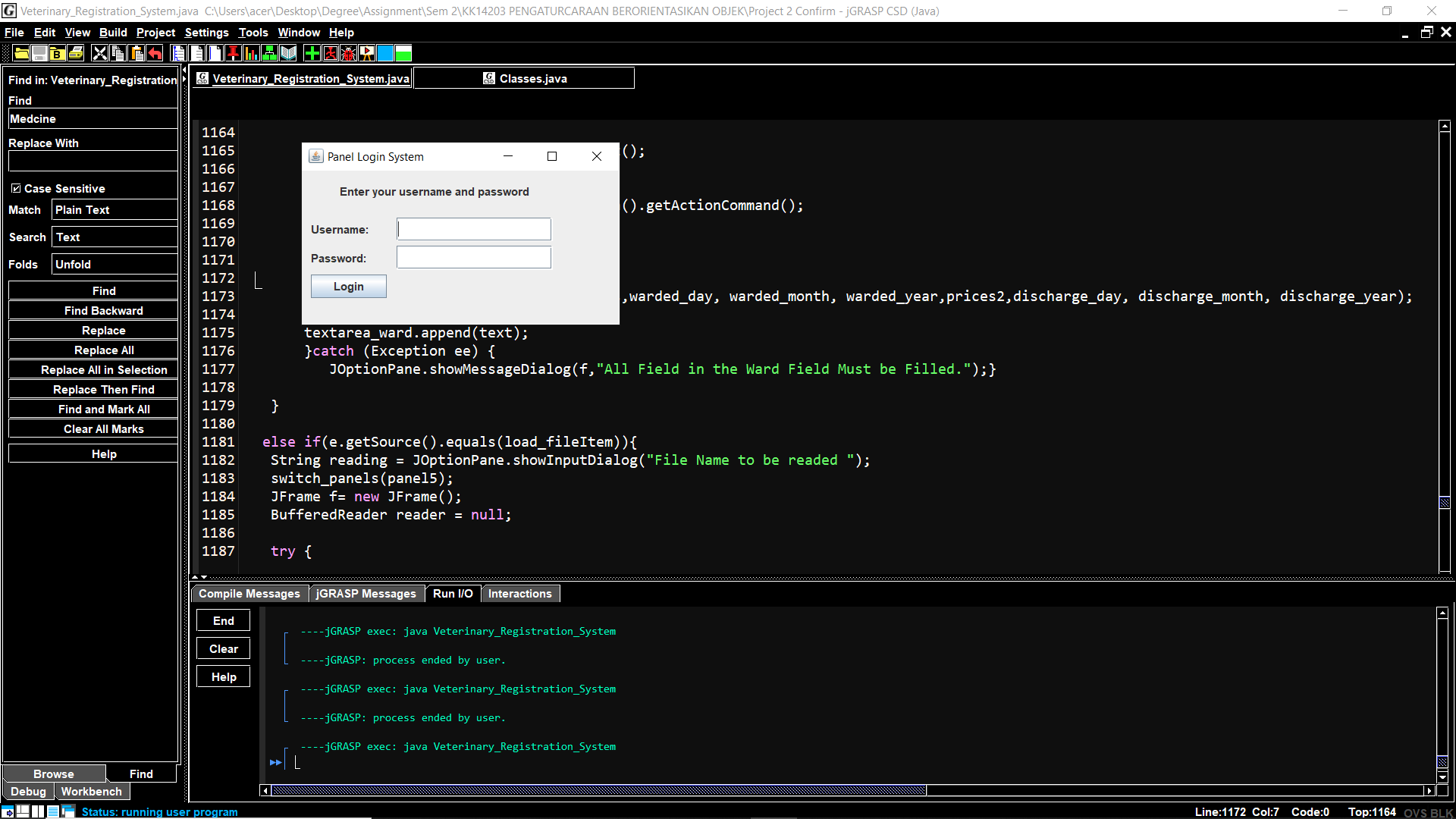
In figure 22, it shows that the information from the figure 20 is successfully writed in a text file named “Hisyam Pet”.

**4.0 User Manual**

In this project, there are 4 panel which is Panel Login System, Choosing Panel, Doctor Biodata Panel and Patient Main Menu Panel.

**4.1 Login Panel**

In this panel, the user require to put in the username and password in order to access the system. The username for this system is “hisyam” and the password is “bi19110028”. If the user fail to put it in the first time, the system will automatically closed due to safety reason.

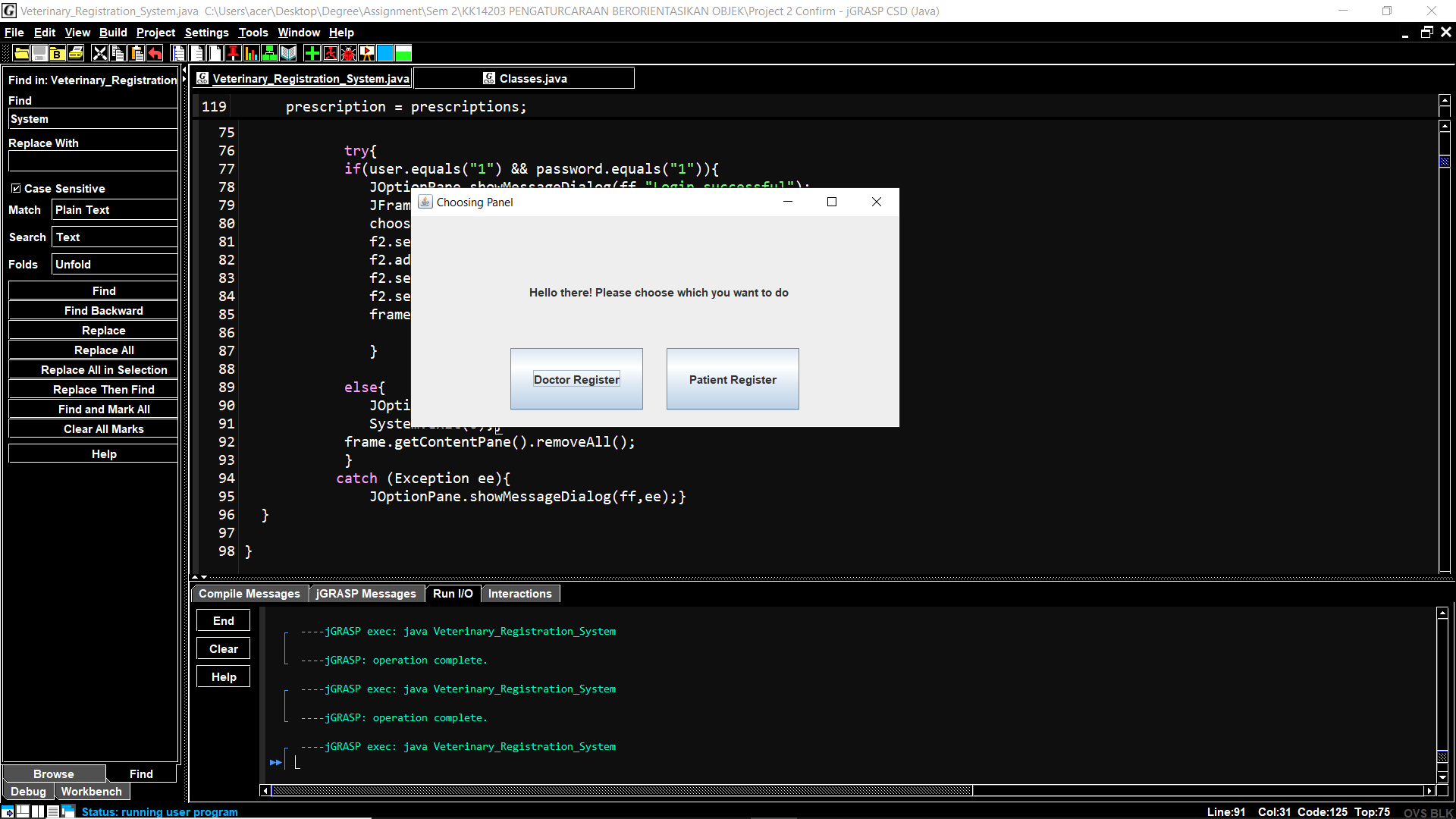


**Figure 23:** Overview of Login panel.

After the user input the username and th password, a message dialogue will appear and indicates that it is successful and the user is transferred to the Choosing Panel.

**4.2 Choosing Panel**

After the user enter the correct password, the user will be entered to the hoosing panel. In this panel, the user require to choose wether to register a new doctor or a new patient.

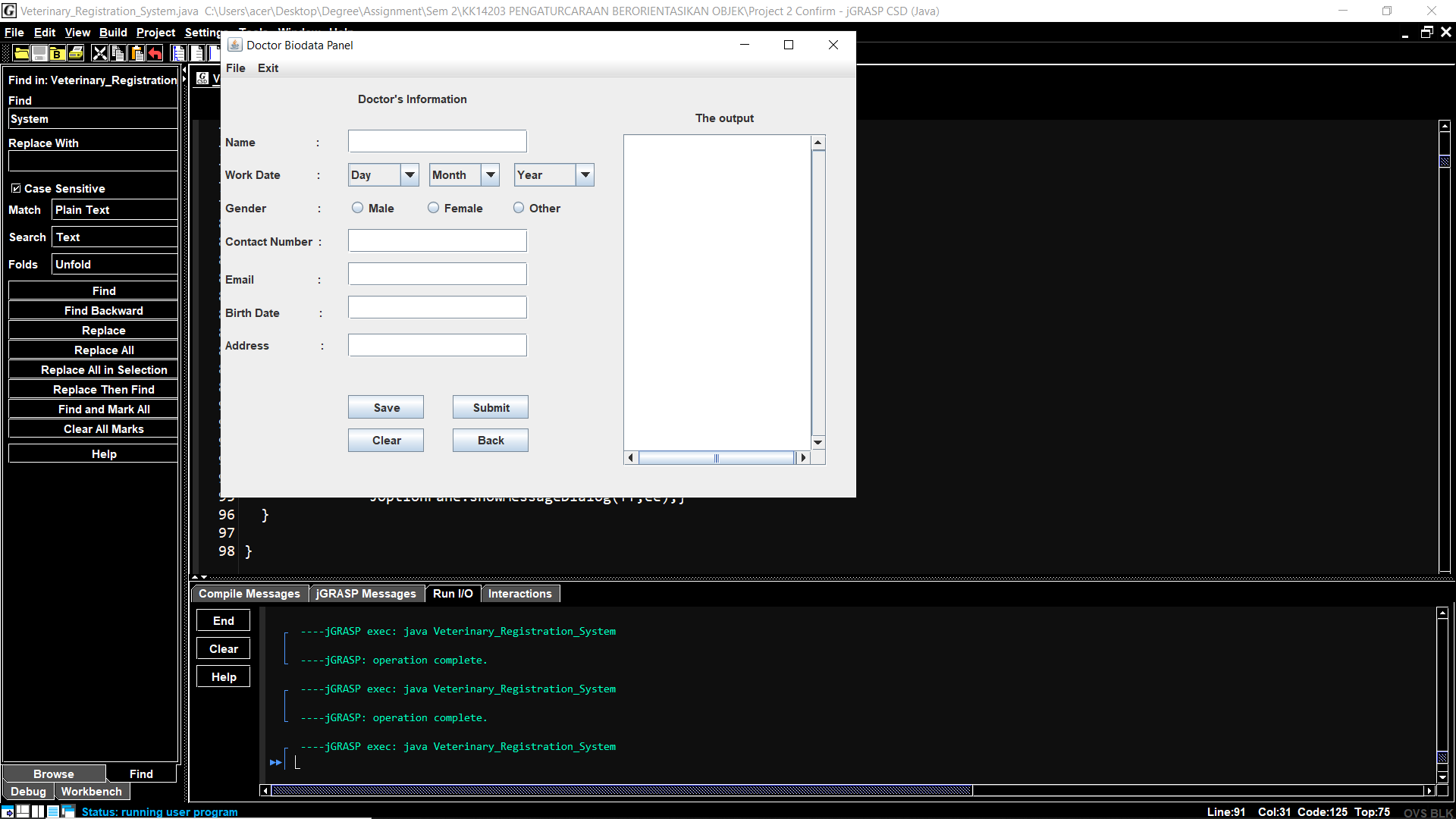


**Figure 24:** Overview of Choosing Panel panel.

There will be two button showing “Doctor Register” and “Patient Register”. The Doctor Register button is to register a new doctor meanwhile, the “Patient Register” buton is used to register a new patient. If the user clicked the “Doctor Register” button, the user will enter the Doctor Biodata Panel. In the same time, if the user clicked the “Patient Register” button, the user will enter the Patient Main Menu Panel.

**4.3 Doctor Biodata Panel**

After the the user clicked the “Doctor Register” button in the choosing panel, the user will enter the Doctor Biodata Panel. In this panel, the user need to fill the doctor biodata.

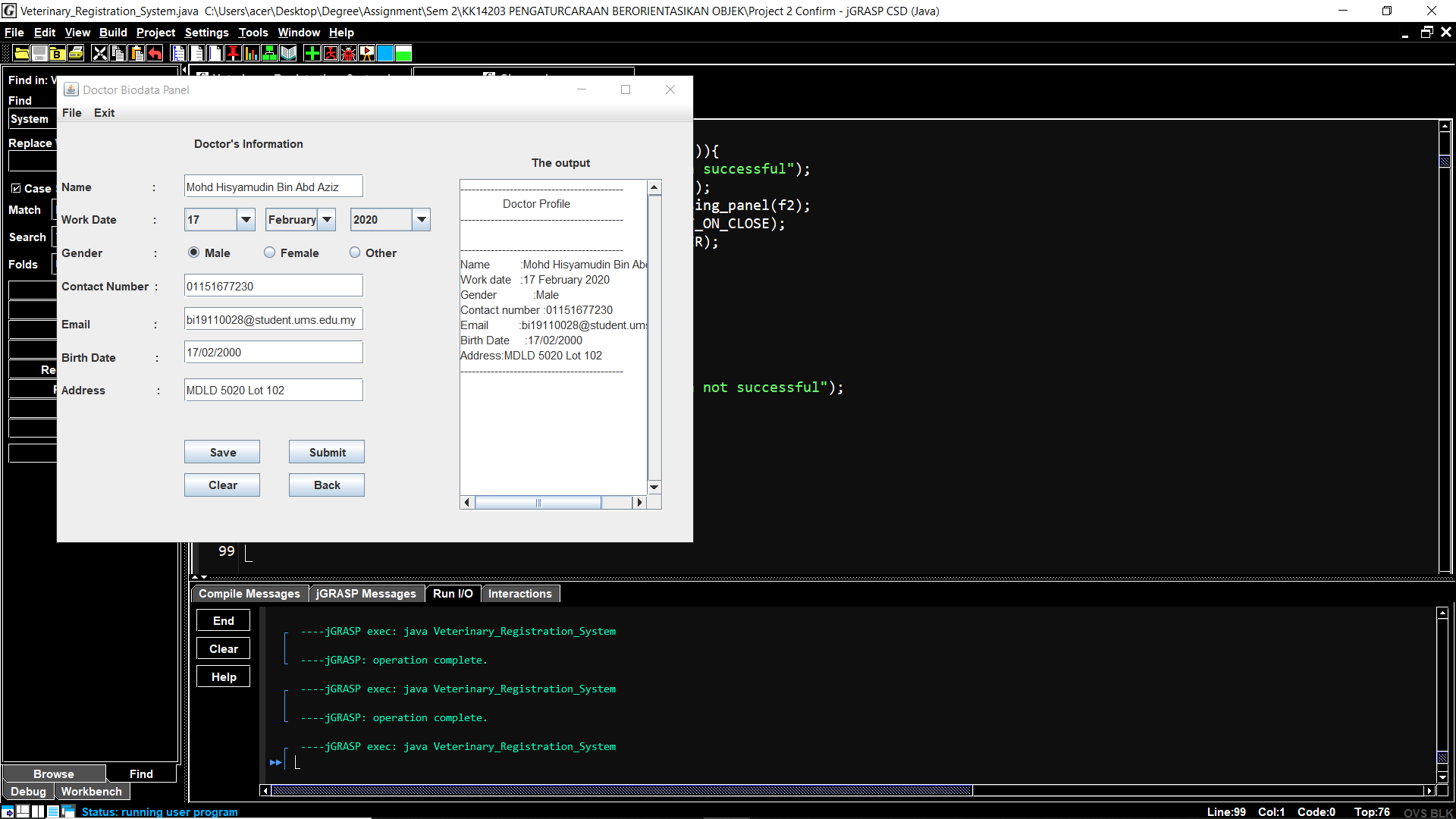


**Figure 25:** Overview of Doctor Biodata Panel panel.

In Doctor Biodata Panel, there are five JtextField, 3 JcomboBox and 3 JRadioButton that need to be filled by the user itself. After that,there are 1 JTextArea to show the output of the input from the user. After that, there are 4 JButton which is “Save”, “Submit”, “Clear” and “Back”. After that, in the Contact Number text field, only number can be input from the user or else there will be an error shown.

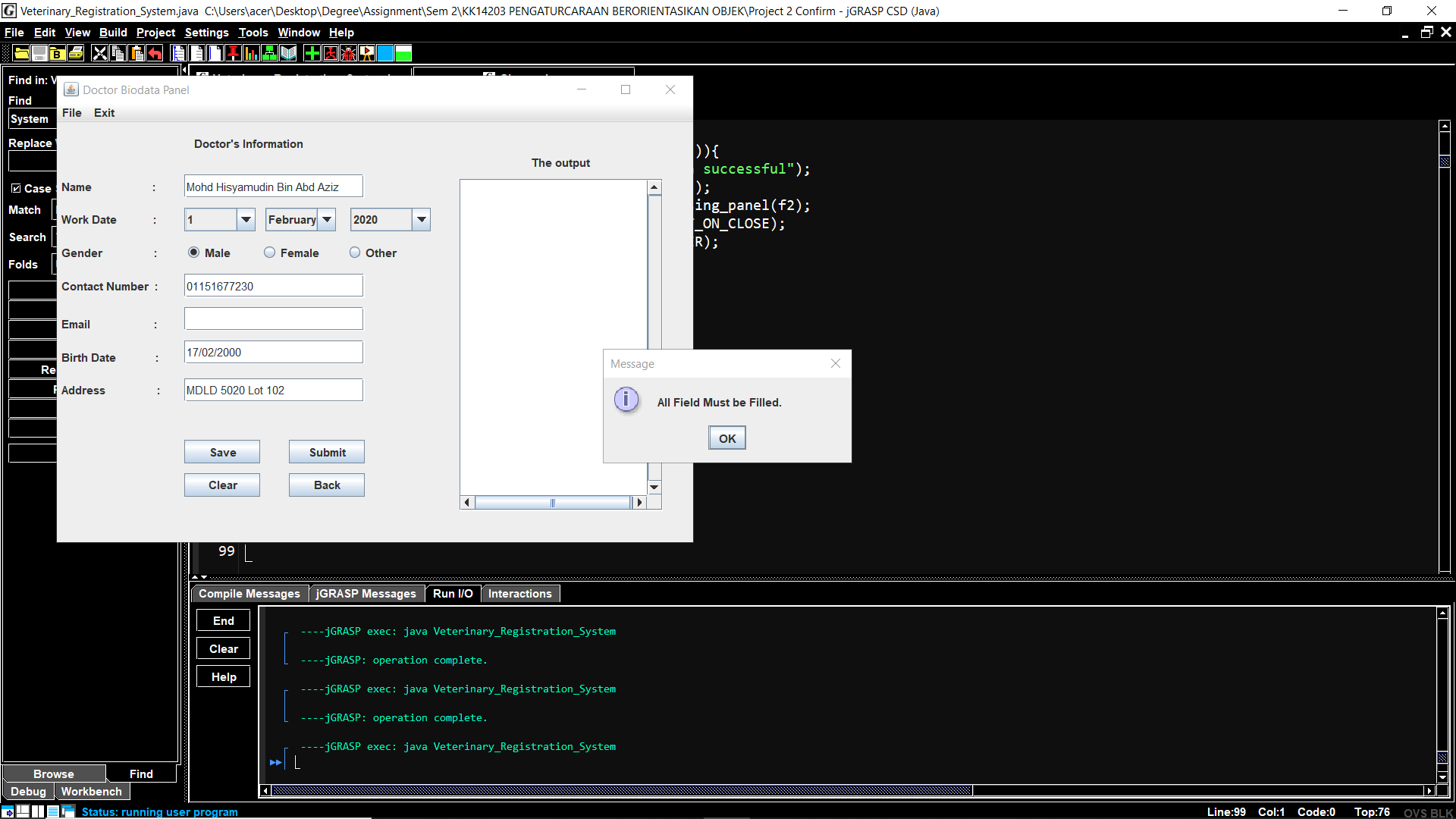
**4.3.1** **Submit Button.**

If the user clicked the “Submit” button, the ouput will be shown into the JtextArea.



**Figure 26:** The ouput is shown in the JTextArea.

However, if there is any field that are empty, there will be no output shown and there will be a message dialogue appear that says “All Field Must be Filled”.

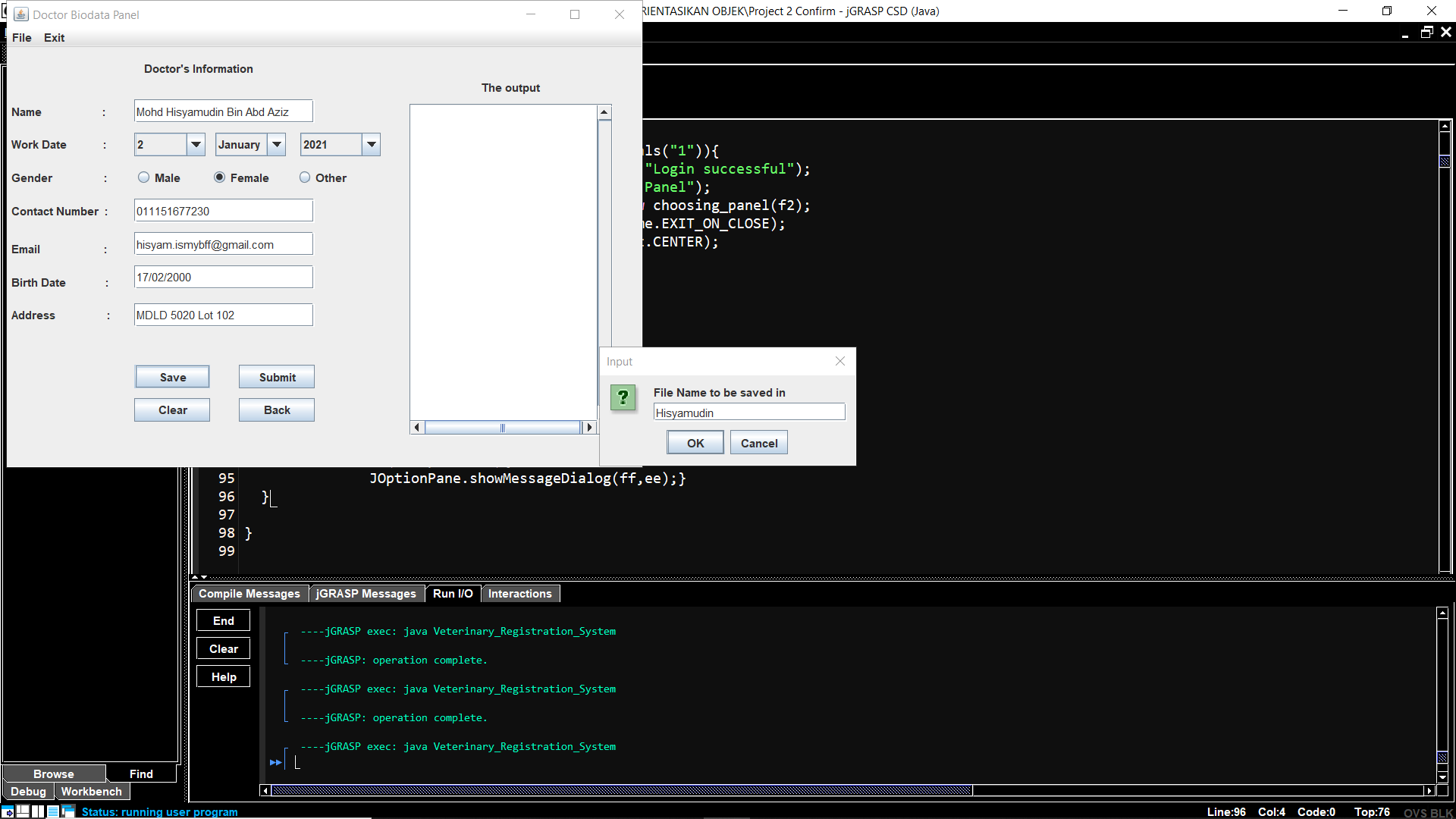


**Figure 27:** A message dialogue will appear if any of the fields are empty.

In figure 27, it shows that the Email JtextField are empty. This will make the JtextArea will not append any output and a message dialogue will appear after the “Submit” button is clicked to make sure the user enter all the field.

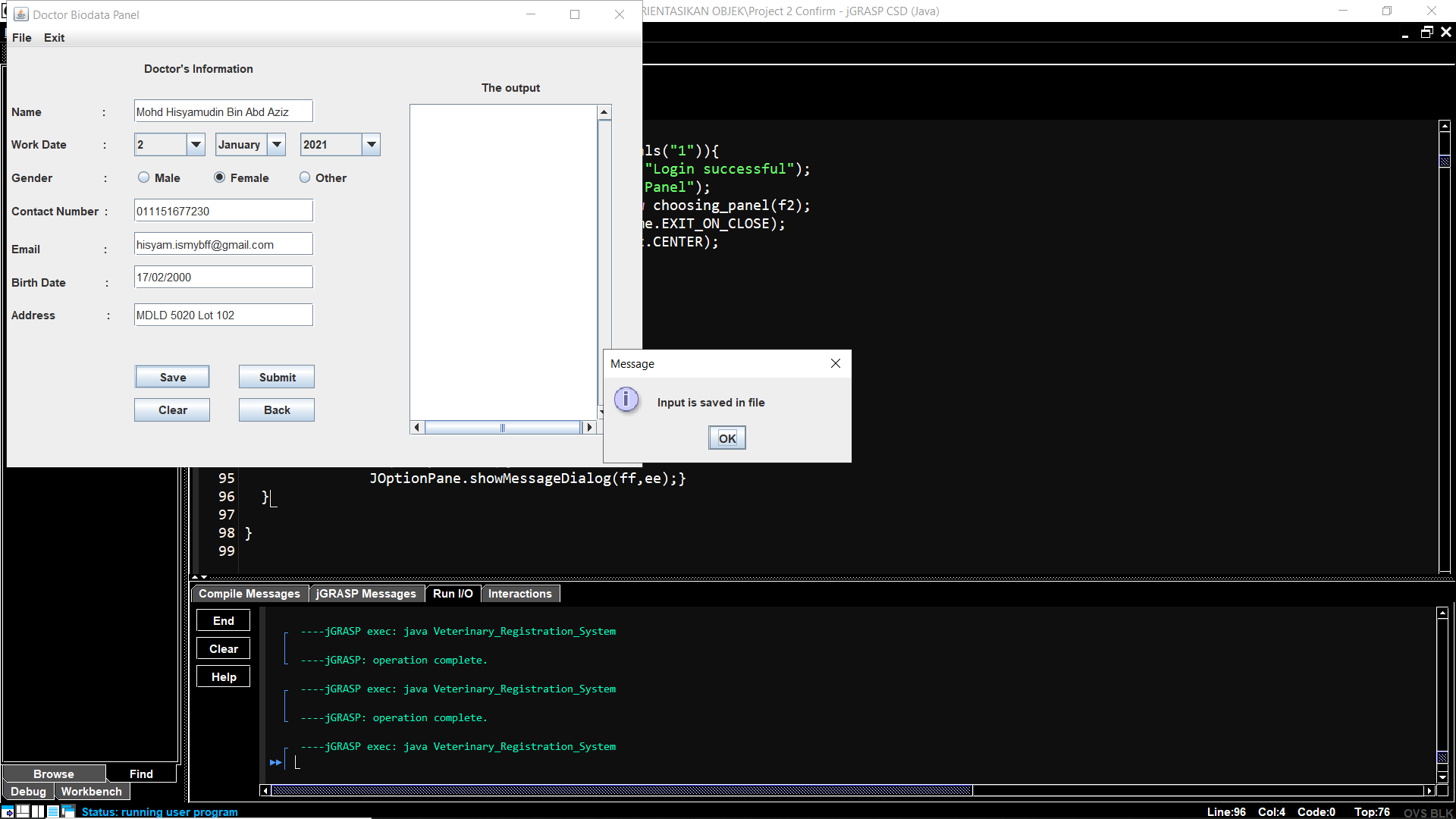
**4.3.2 Save Button.**

If the user clicked the “Save” button after all fields are filled, an input dialogue will appear. It is for the user to put the name of the file where the data need to be write in.



**Figure 28:** An input dialogue will appear after the user clicked “save” button.

Figure 28 shows that there will be an input dialogue appear to let the user input the file name where they desired the data to be saved.

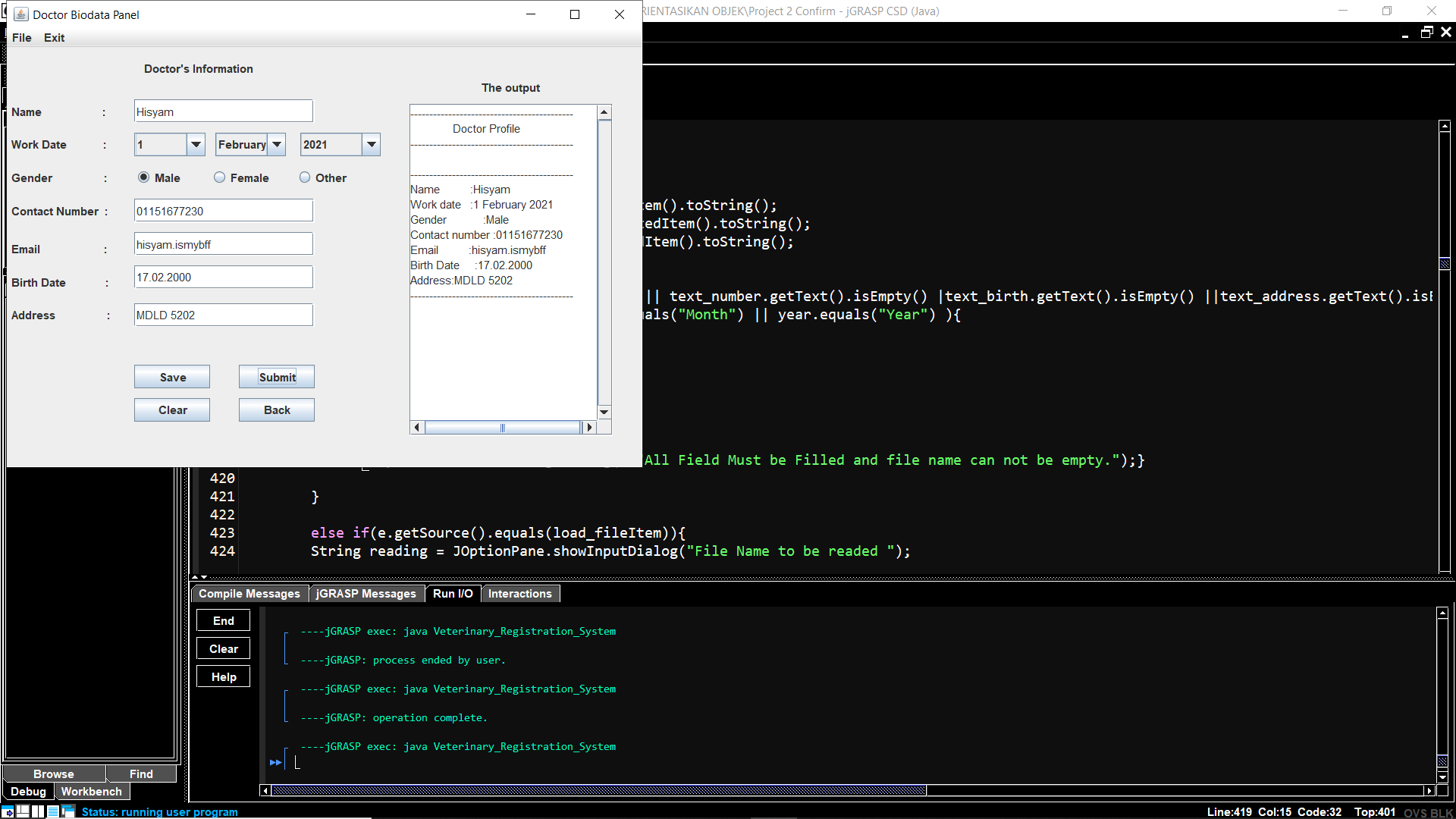


**Figure 29:** A message dialogue will appear if the message is successfully saved.

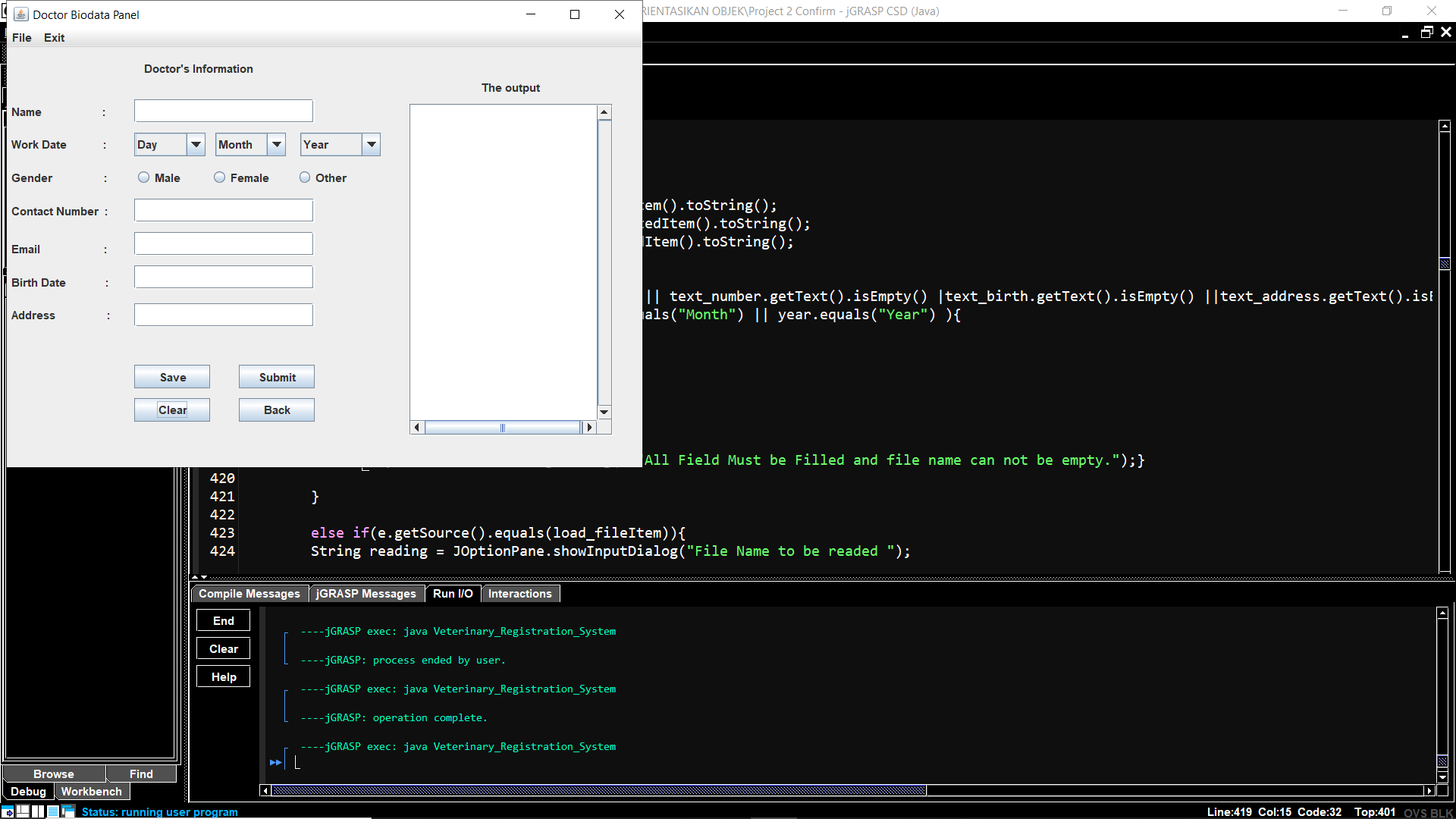
In figure 29, it shows that, if the file is succesfully created and writed the data, a message dialogue will appear indicating that it is succesful.

**4.3.3 Clear Button.**

If the user clicked the clear button, all of the JtextField will be cleared including the JtextArea.



**Figure 30:** Before the clear button clicked.

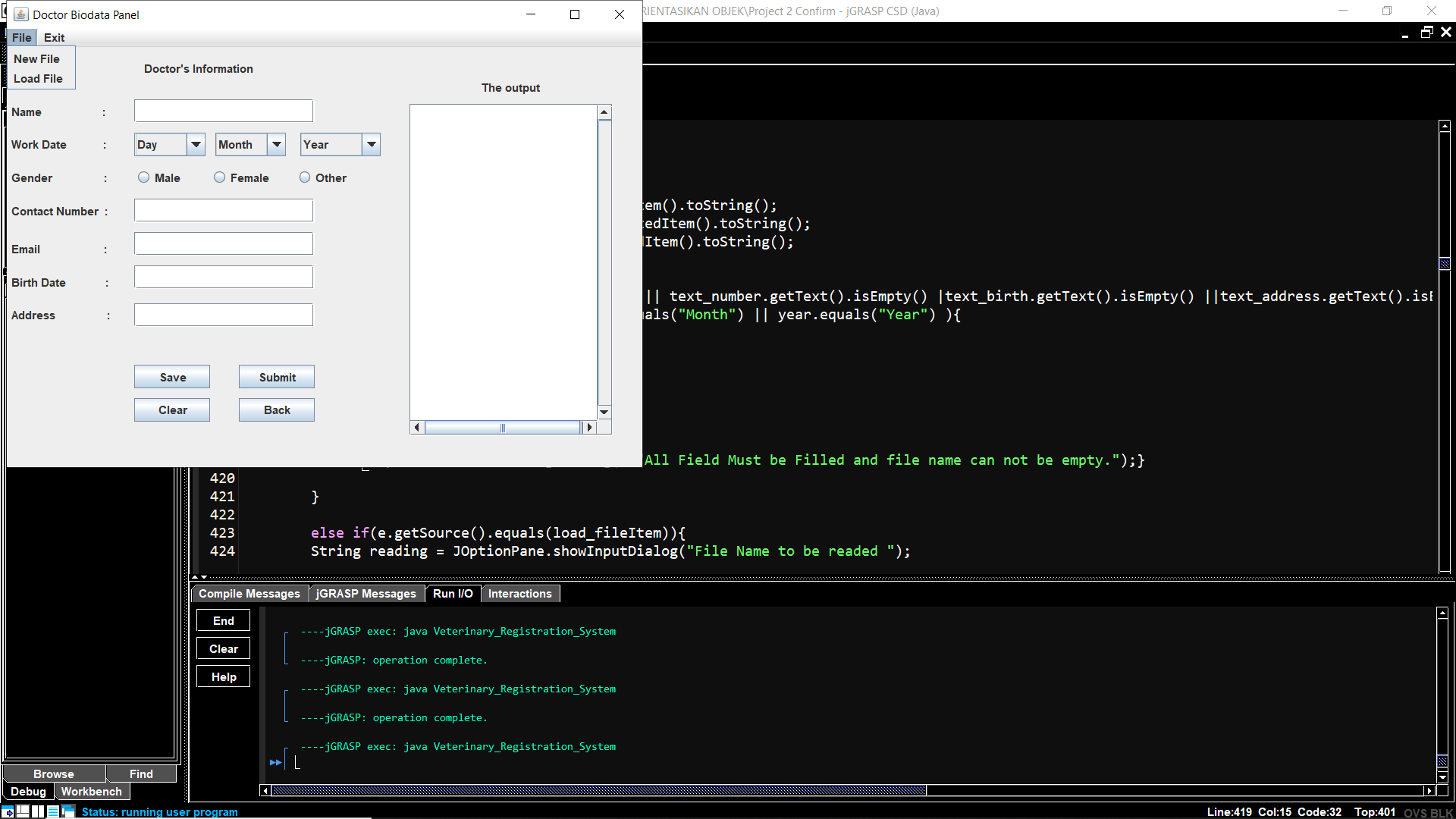


**Figure 31:** After the clear button clicked.

**4.3.4 Back Button.**

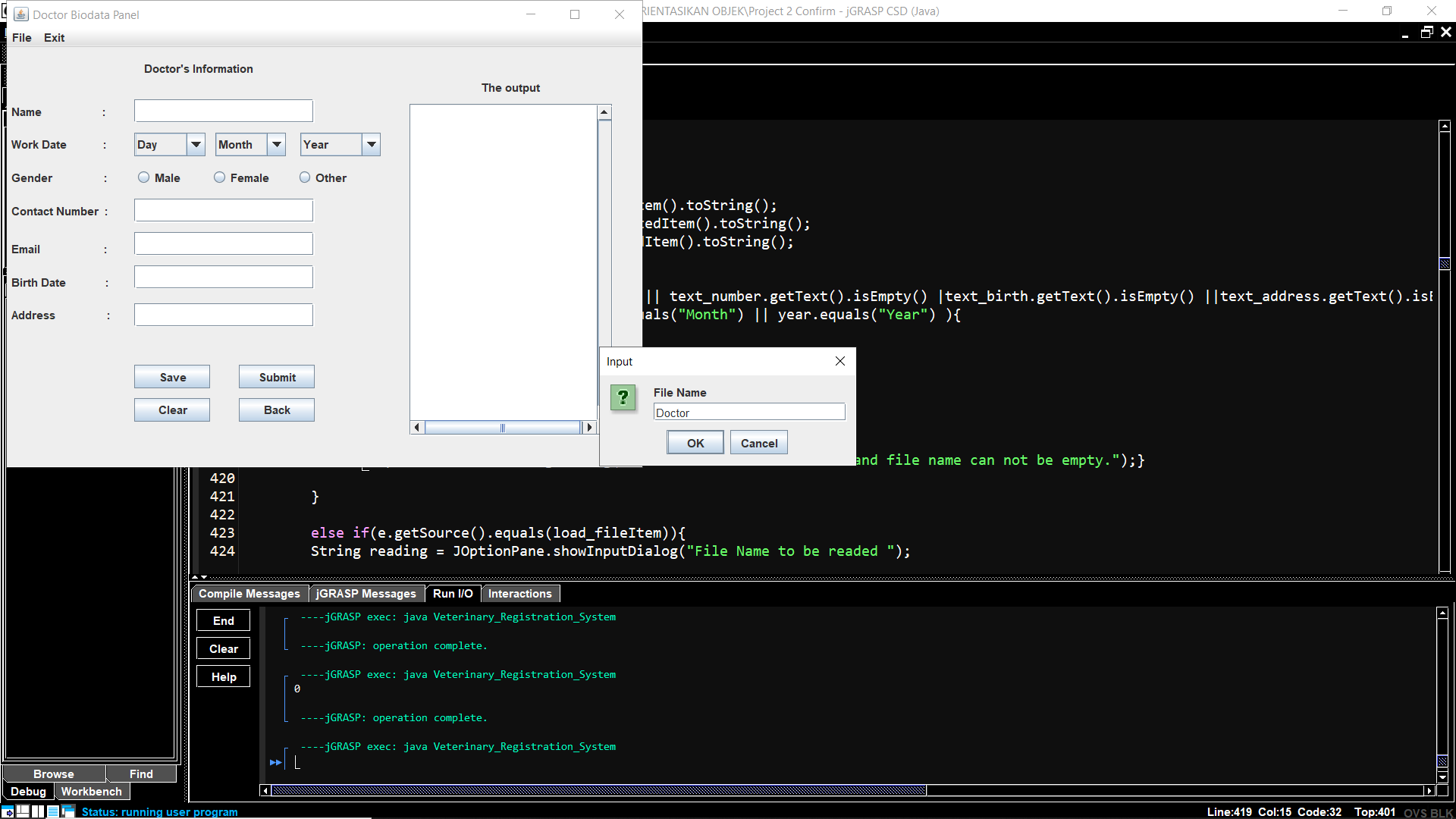
If the user clicked the “Back” button, the user will be brought to the Choosing Panel again to choose what the user need to do again.

**4.3.5 Menu and Menu Item**



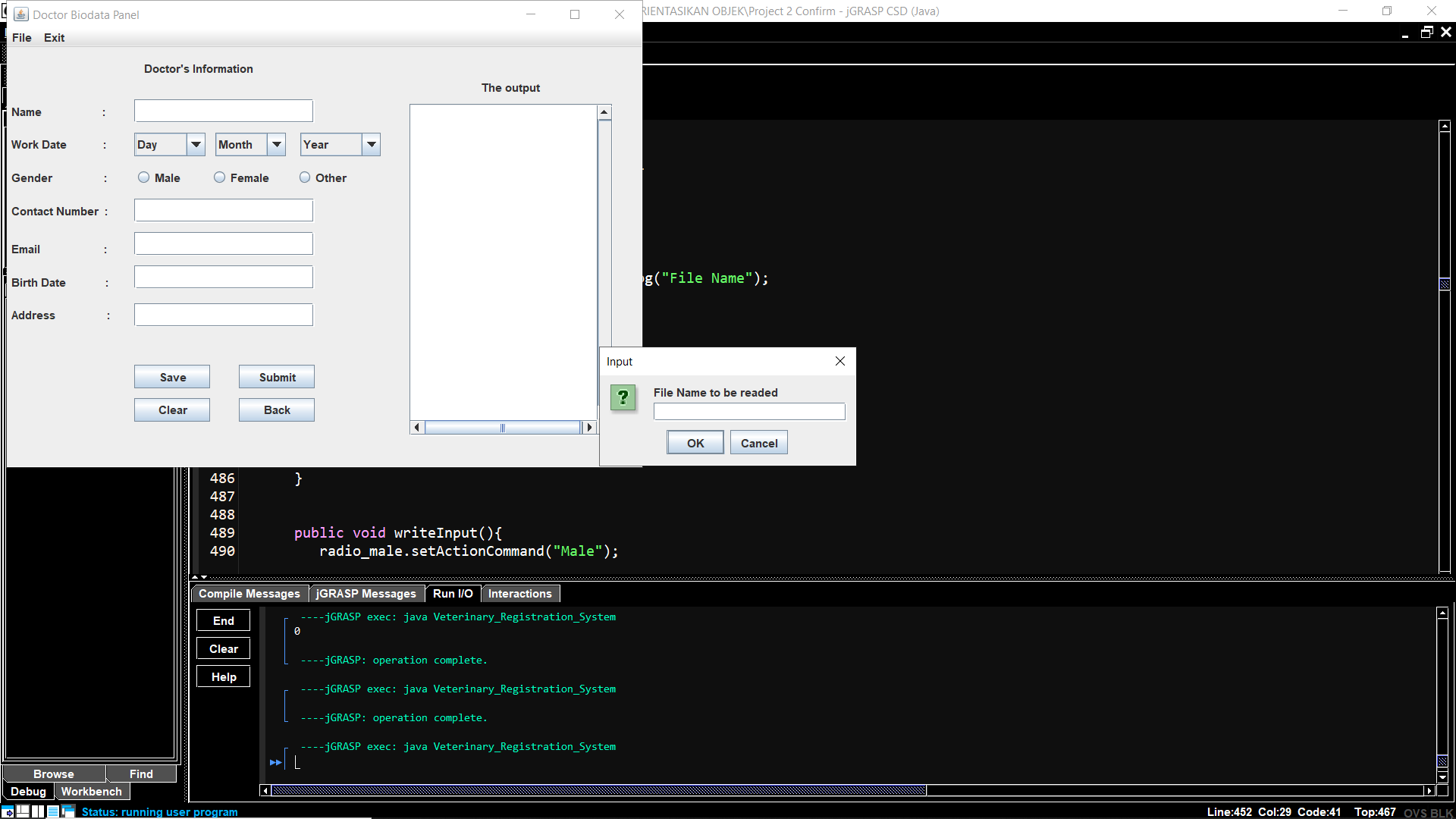
**Figure 32:** Menu and Menu Item in the Doctor Biodata Panel.

In the Doctor Biodata Panel, there is 2 Menu which is File and Exit as in figure 27. After that, in File menu, there is two Menu Item which is New File and Load File. New file is for creating a new file and Load file is to read a file.



**Figure 33:** A file named Doctor To be created.

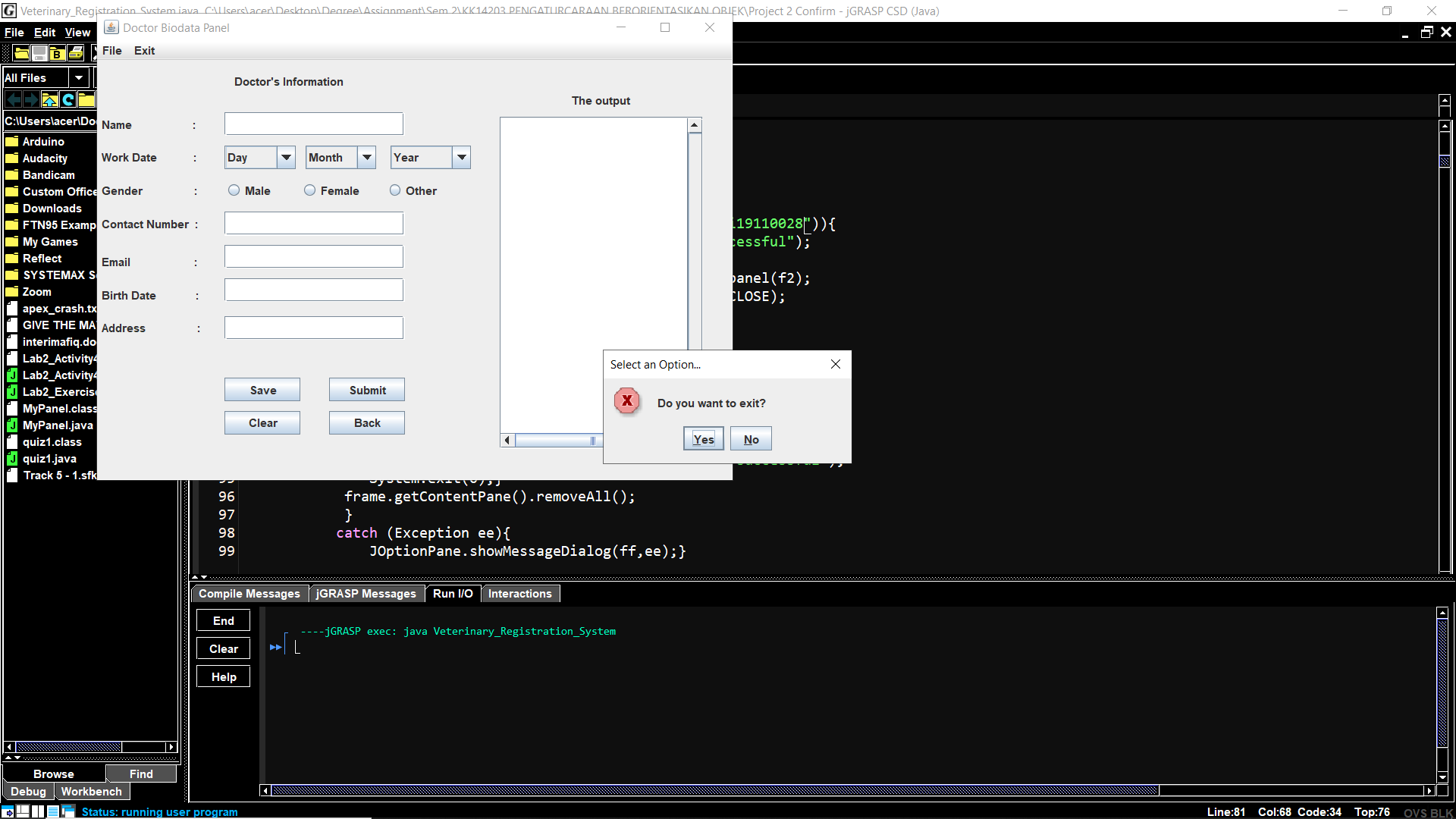
If the user clicked the create file menu item, an input dialogue will appear as in figure 29 and the user need to input a name for the file.



**Figure 34:** The user need to input a file name to be readed.

Meanwhile, if the user clicked Load File, there will be an input dialogue appear as in figure 30 and the user need to input the name for a file to be readed.

After that, if the user clicked the exit menu, an option dialogue will appear to ask the user if they really want to exit.

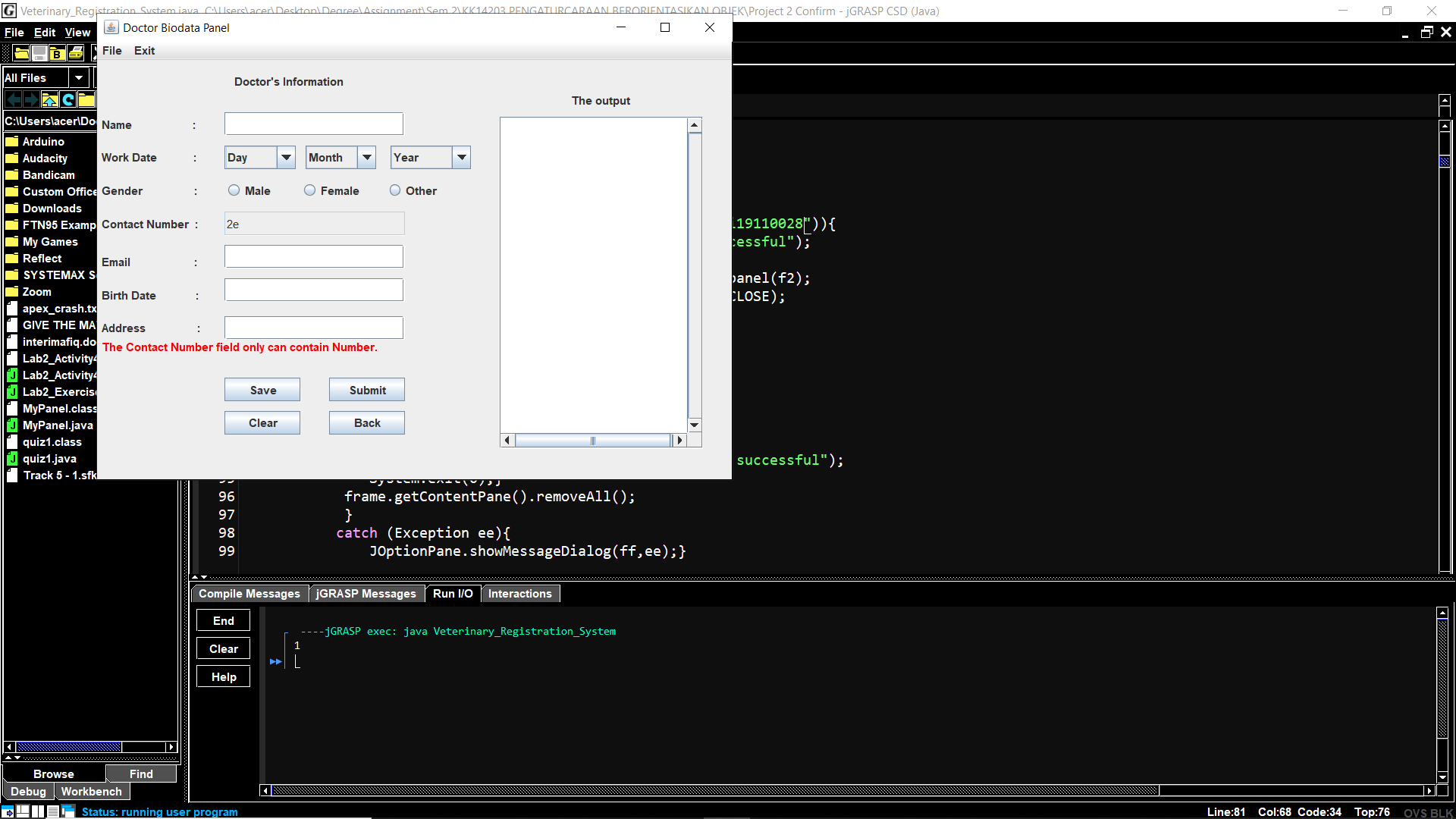


**Figure 35:** An option dialogue will appear after the user clicked exit.

If the user clicked yes, the system will be stop executing.

**4.3.6 Fields that need to be filled**

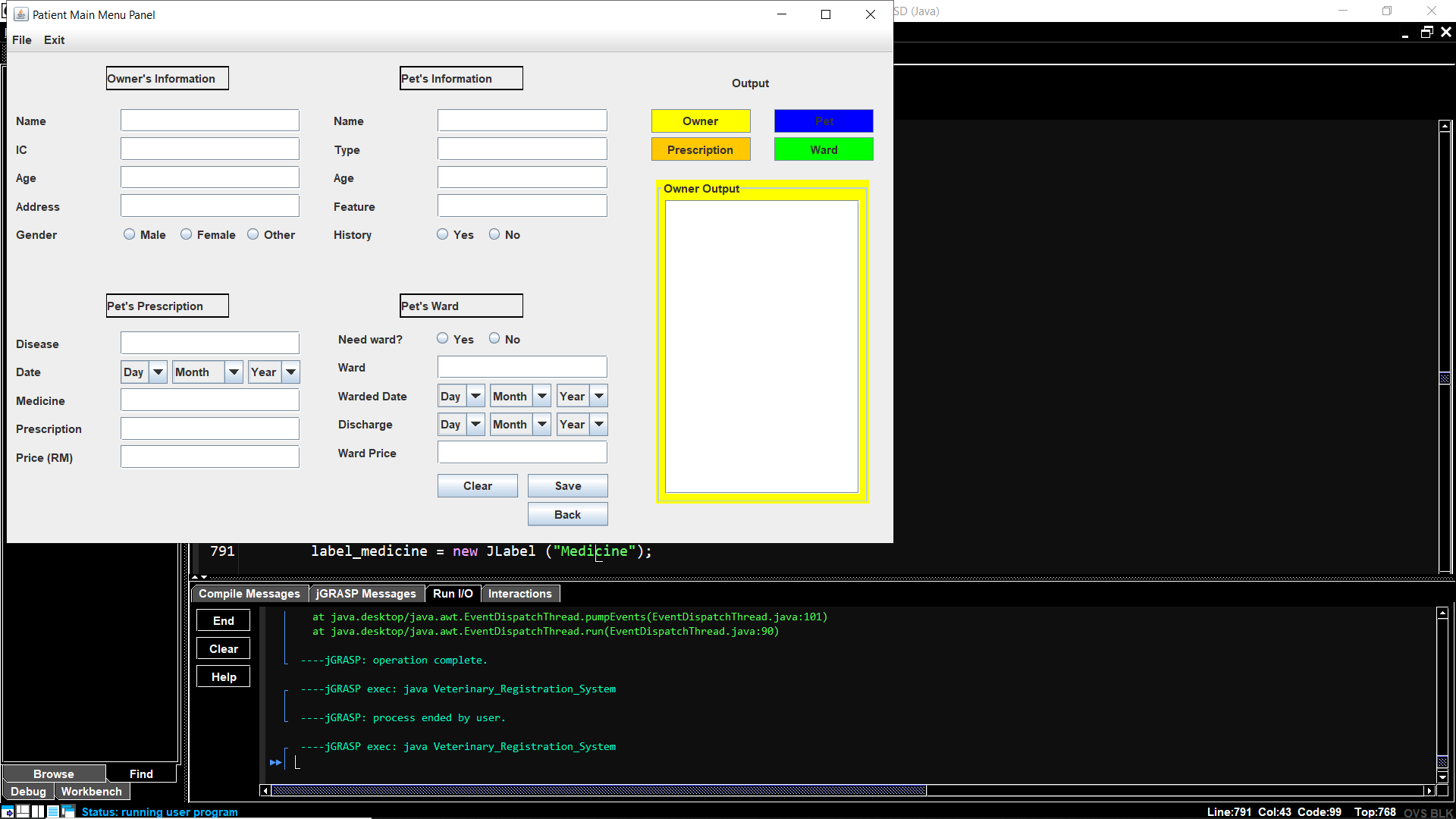
In the system, all of the field must be filled before clicking the submit and save button. The name, work date, gender, email, birth date and address will accept all types of input which means that they are a string type. Meanwhile, the contact number only can contain integer which is number.



**Figure 36:** There is an error shown if the user input a character besides number.

Figure 36 shows that, the user only can input number in the fields. If the user input a letter or any other words besides a number, an error message will appear saying “The Contact Number field only can contain number”. In the same time, the user can not input another character if they do not erase the character besides an integer they can not edit the fields.

**4.4 Patient Main Menu Panel**

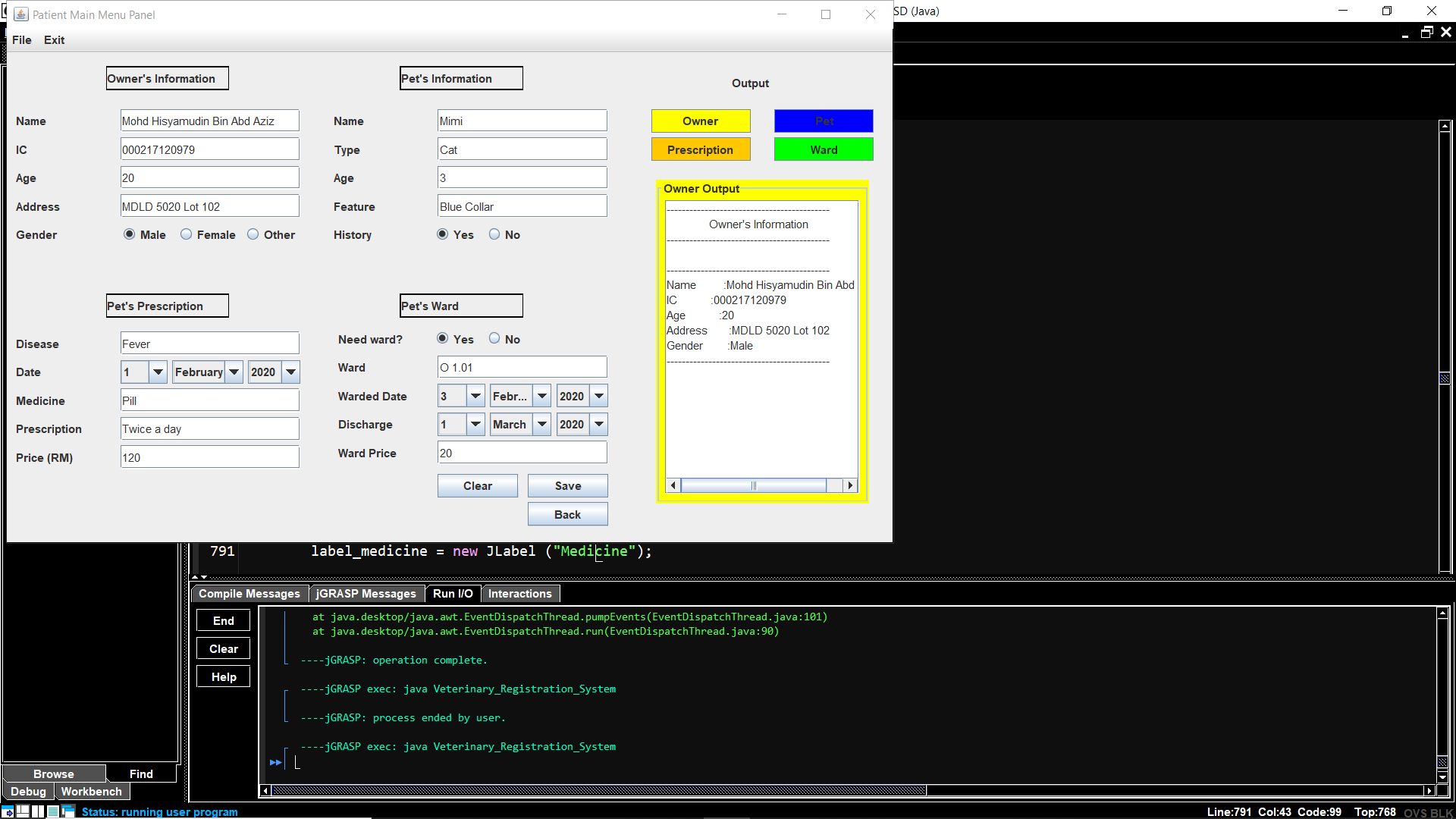


**Figure 37** Overview of the Patient Main Menu Panel.

In Patient Main Menu Panel panel, there are divided by four categories which is Owner’s Information, Pet’s Information, Pet’s Prescription and Pet’s Ward. Each categories have 5 fields that need to be filled by the user. If any of the field is empty, a message will appear that tells the user to fill the empty field. The menu function in this panel are the same with Doctor Biodata Panel. The button “Save”, “Back” and “Clear” also have the same function as in Doctor Biodata Panel.

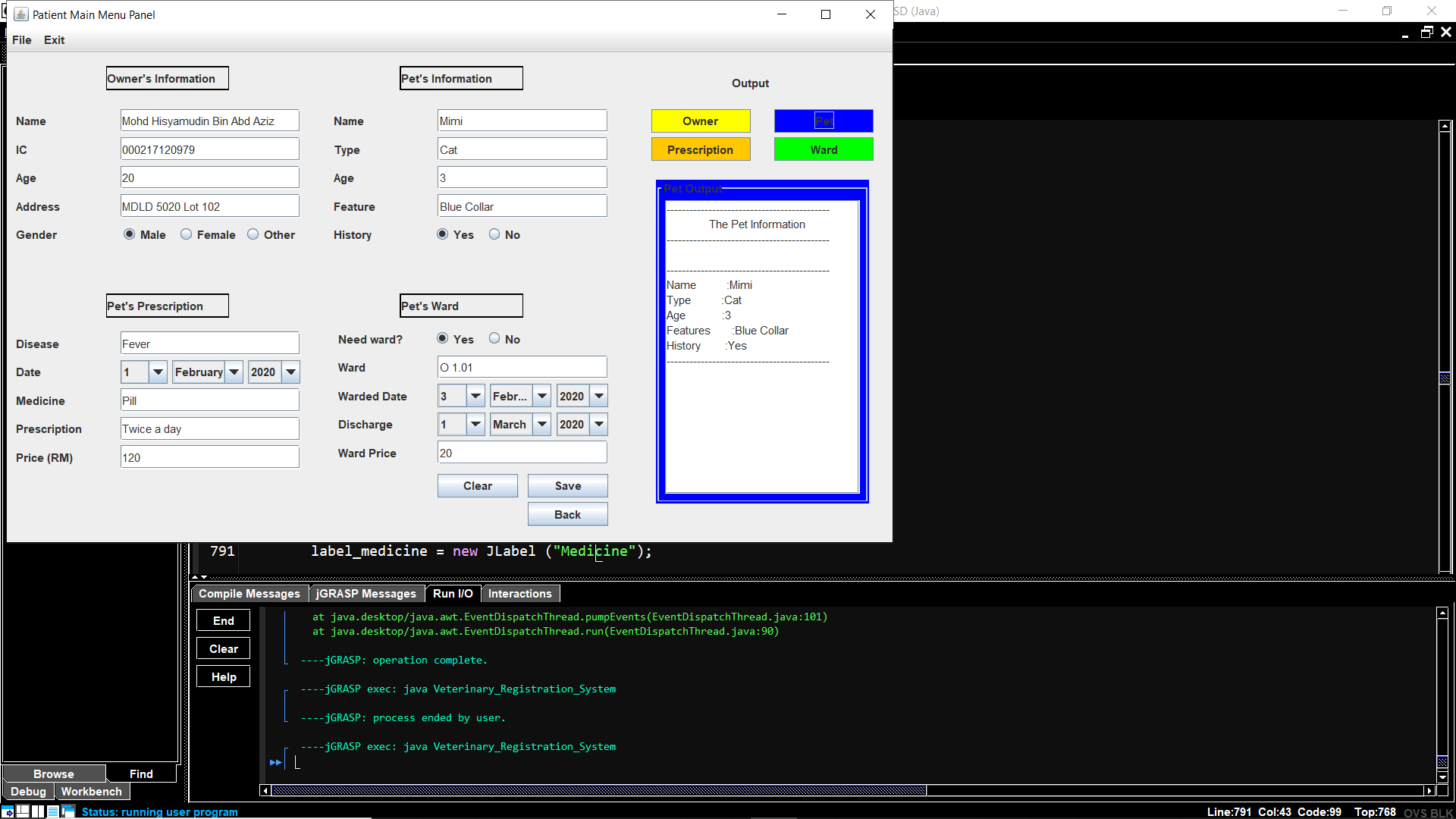
**4.4.1 Owner, Pet, Prescription And Ward Button.**

Each of the button is to put output of the categories to the JtextArea. The colour of the button indicates the output panel that is currently shown. For an example, in figure 31, the JtextArea Owner’s Ouput colour is yellow and the Jbutton is also yellow in colour.



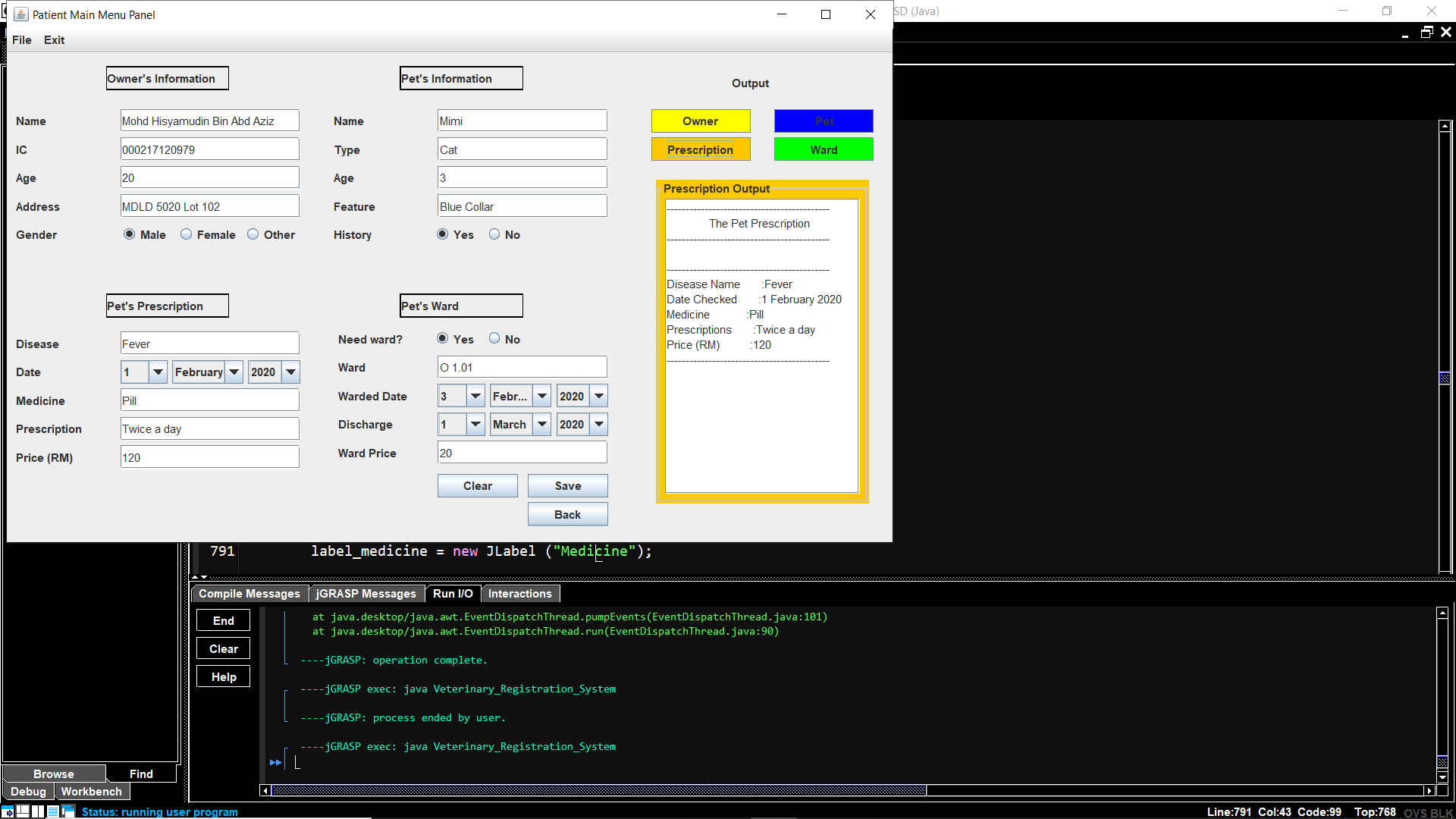
**Figure 38:** Owner’s output is shown.

In figure 38, the Owner’s output is shown and is yellow in colour. The colour can help to differentiate who output is currently shown in the screen right now.

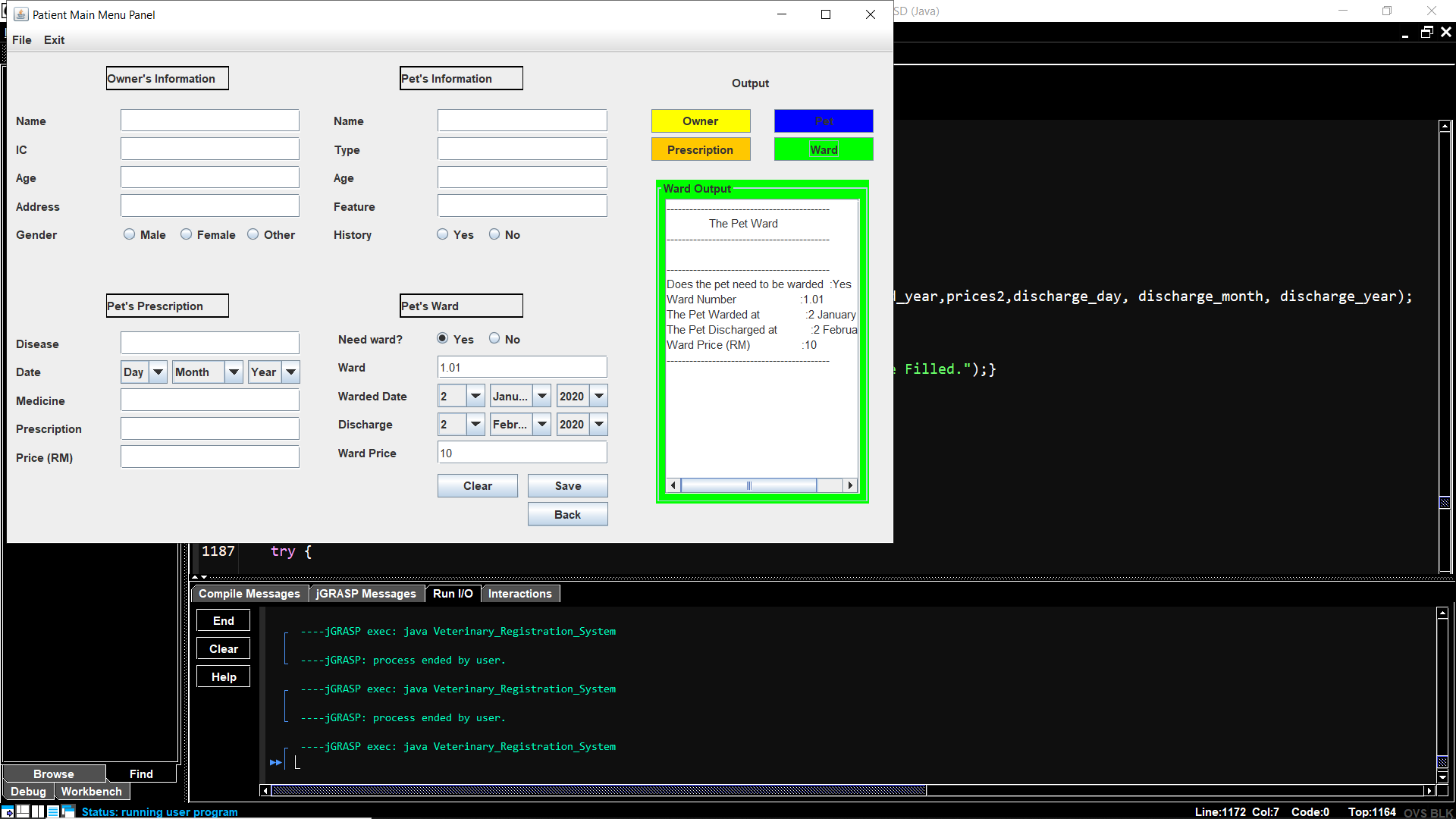


**Figure 39:** Pet’s output is shown.

In figure 39, the Pet’s output is shown and is blue in colour.

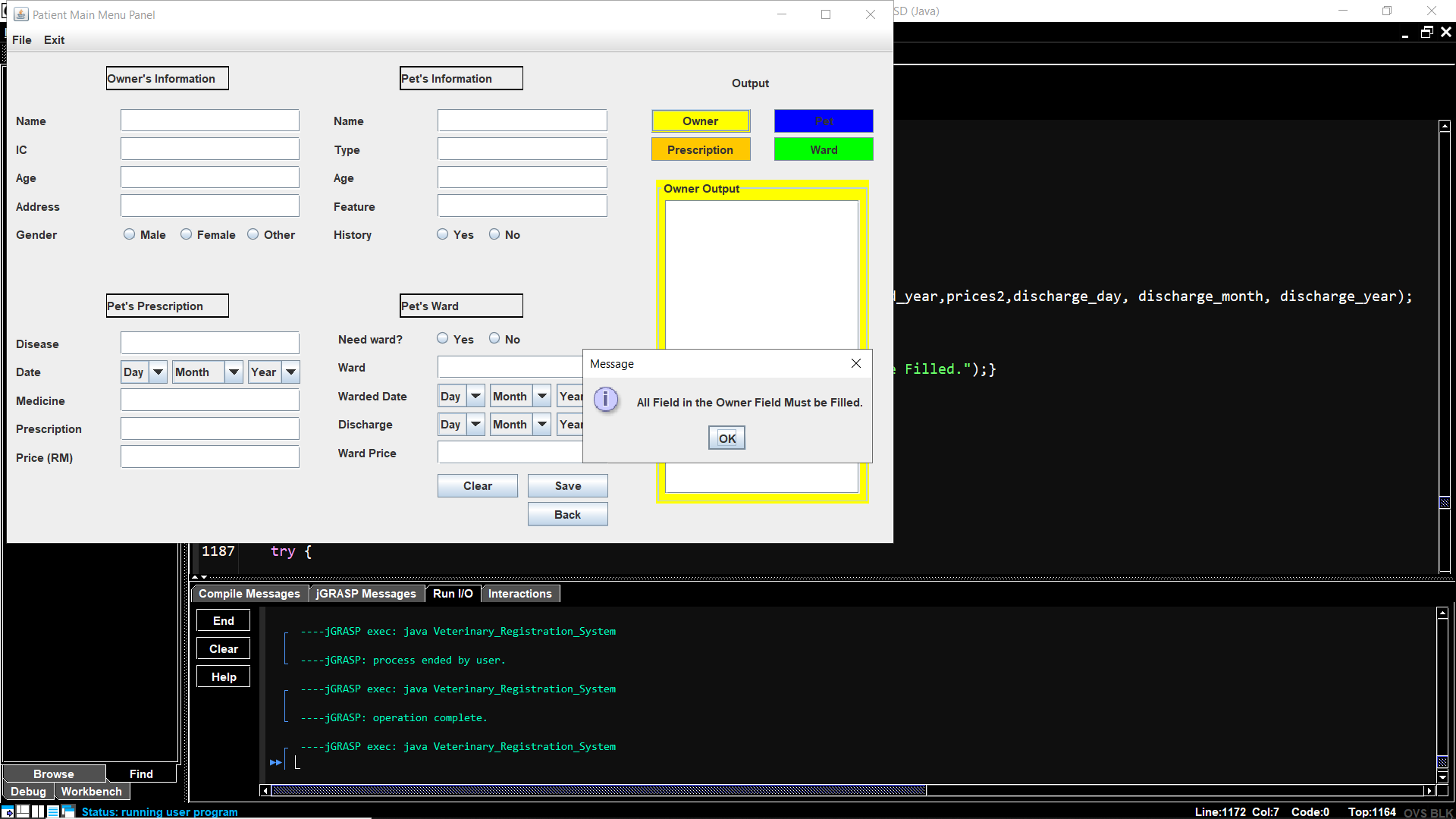
 **Figure 40:** Prescription’s output is shown.

In figure 40, the Prescription’s output is shown and is orange in colour.



**Figure 41:** Ward’s output is shown.

In figure 41, the Ward’s output is shown and is green in colour.

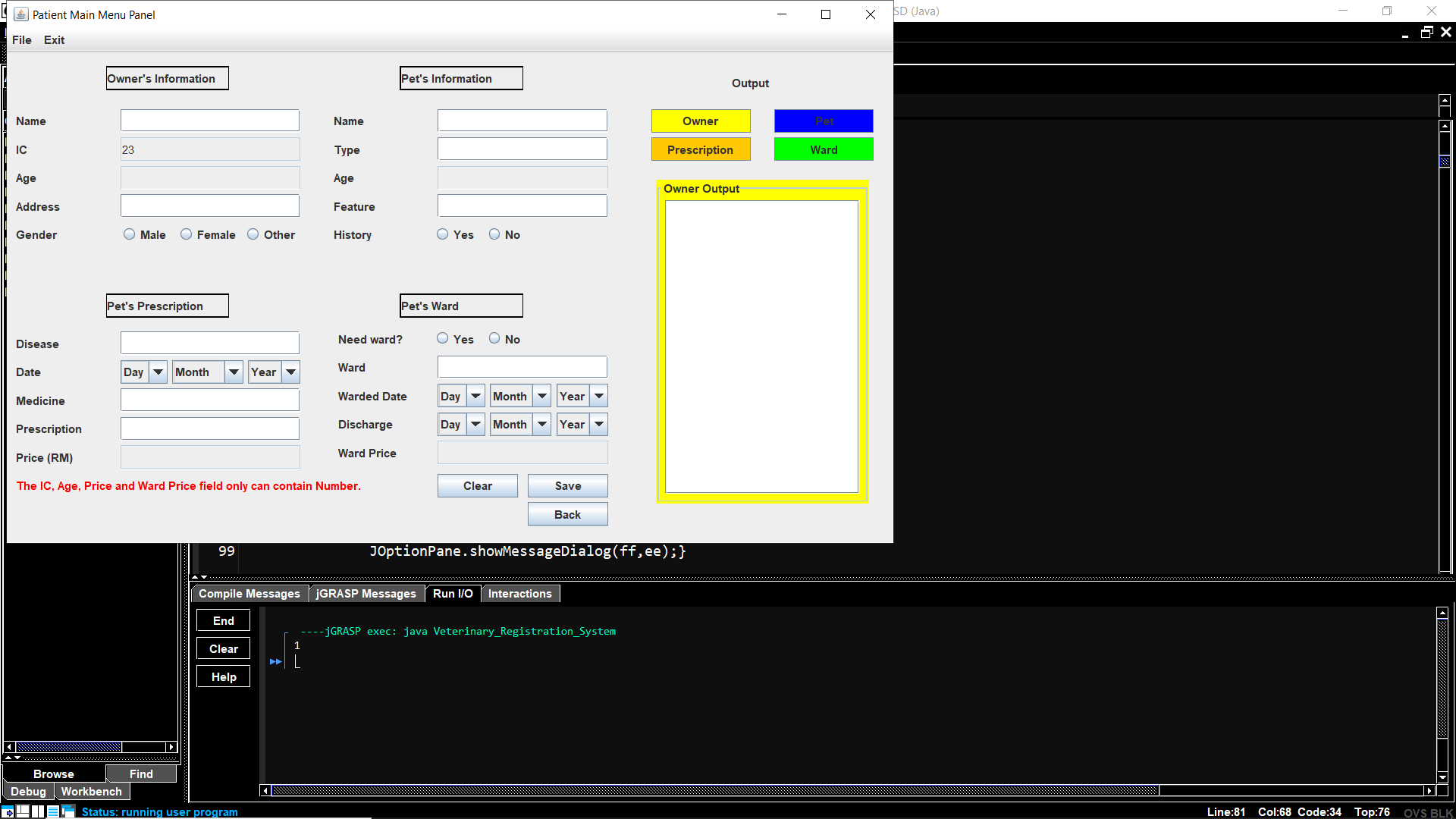


**Figure 42:** Message dialogue appears if any of the text field does not fill within their categories.

In figure 42, it shows that a message dialogue sayaing “All Field in the Owner Field Must be Fill” will appears if any of the text field does not fill within their categories. So to avoid this, the user need to fill in all the information to print the ouput in the text area.

**4.4.2 Fields that need to be filled**

In the Patient Main Menu Panel, all of the field must be filled before clicking the Owner, Pet, prescription, Ward and Save button. The Name , address, gender, Type, Feature, History, Disease, Date, Prescription, Ward, Warded Date and Discharge will accept all types of input which means that they are a string type. Meanwhile, the IC, Age, Price(RM) and Ward Price only can contain integer which is number.



**Figure 43:** There is an error shown if the user input a character besides number.

Figure 43 shows that, the user only can input number in the the IC, Age, Price(RM) and Ward Price fields. If the user input a letter or any other words besides a number, an error message will appear saying “The IC, Age, Price and Ward Price field only can contain Number”. In the same time, the user can not input another character if they do not erase the character besides an integer they can not edit the fields.