
# Object Oriented Programming Report Documentation

**TECHNOLOGY PARK MALAYSIA**

**Object Oriented Programming (AAPP013-4-2-OOP-L-1)**

**UCDF2304ICT(SE)**

**Group Thick of It**

**Member 1:** Wong Kap Onn (TP074292)

**Member 2:** Tan Yi Han (TP070378)

**HAND OUT DATE:** **Wednesday, 1st August 2024**

**HAND IN DATE:** **Monday, 14th October 2024 @ 11:59PM**

**WEIGHTAGE:** **100%**

---

**INSTRUCTIONS TO CANDIDATES:**

1. Submit your assignment online in Moodle Folder unless advised otherwise

2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld

3. Cases of plagiarism will be penalized

4. You must obtain at least 50% in each component to pass this module

# Table of Contents

# 1.0 Introduction

The Department of Health in Selangor has employed the "Thick of It" group to program a Personal Protective Equipment (PPE) Inventory Management System which is used to manage the inventory of PPEs. The PPEs can either be received from multiple suppliers or distributed to hospitals. The Personal Protective Equipment (PPE) Inventory Management System must be able to manage user information, create inventory for PPEs, manage item inventory, update item inventory, track item inventory, manage supplier information, manage hospital information and produce search functionalities. Text files are used for storing and retrieving data required for the system.

The inventory system is produced in Java Language and with Graphical User Interface (GUI). Object-oriented programming concepts are used for the program.

# 2.0 Design Description & Justification

Login

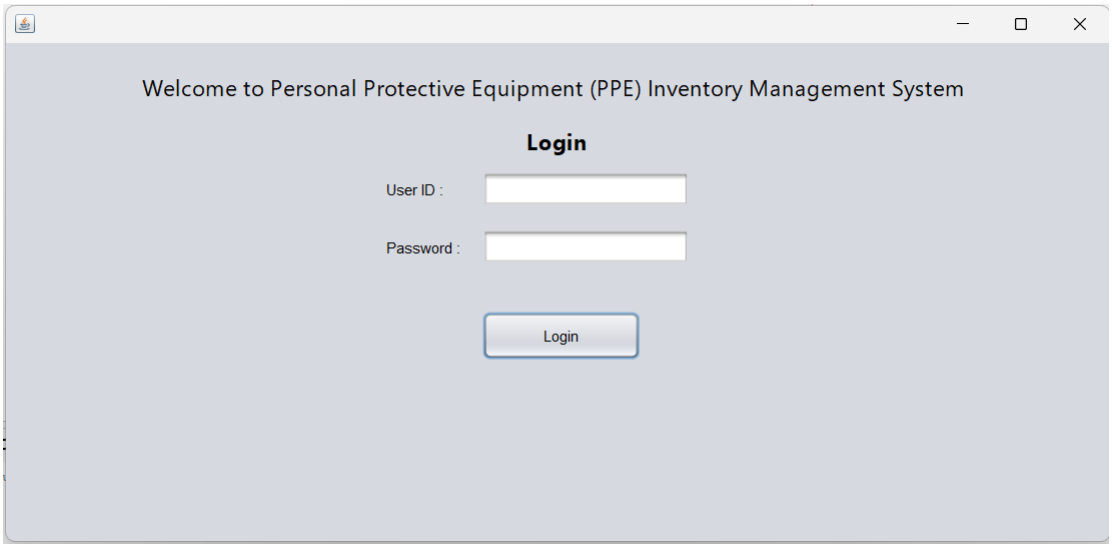



Diagram 1: Login

This is the Login JFrame in the PPE Inventory Management System, where user needs to login using user id and password to access the system. If user id and password matches is correct, then user is granted access to the system.

Admin Homepage

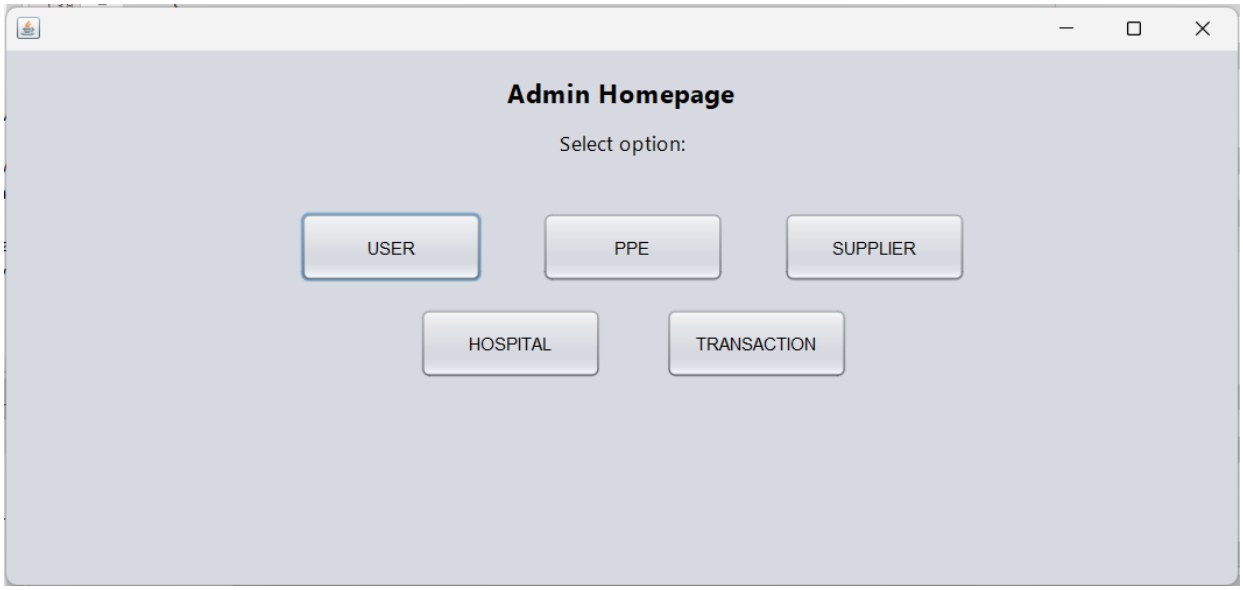



Diagram 2: Admin Homepage

This is the Admin Homepage JFrame in the PPE Inventory Management System, where admin can select which data to access. User will be directed to this JFrame after successful login and if user type is admin.

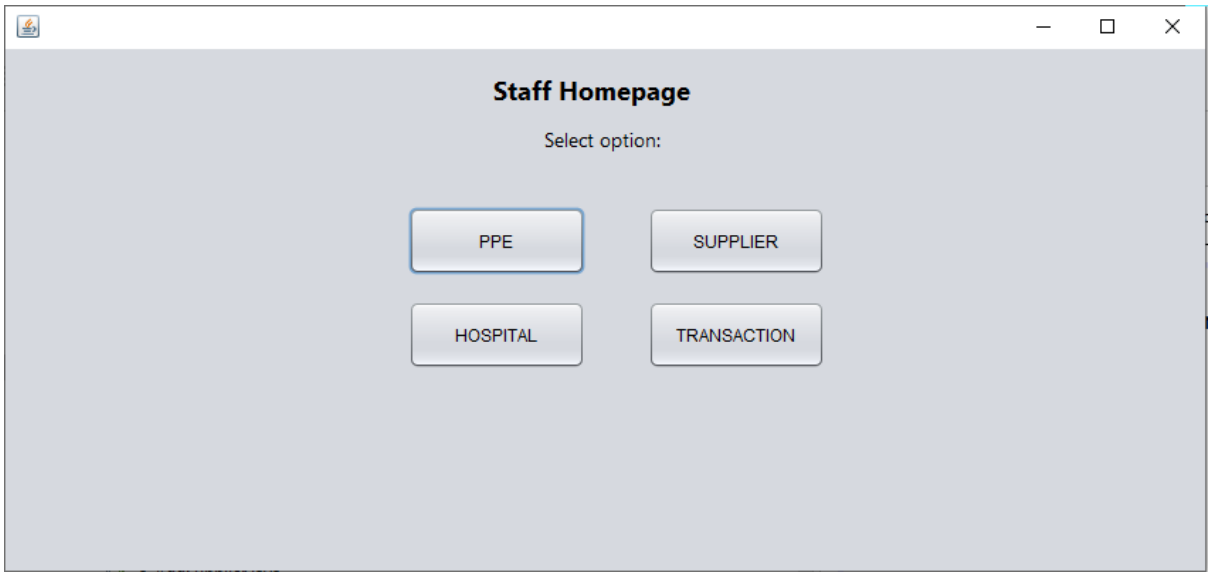



Diagram 3: Staff Homepage

This is the Staff Homepage JFrame in the PPE Inventory Management System, where staff can select which data to access. User will be directed to this JFrame after successful login and if user type is staff.

User

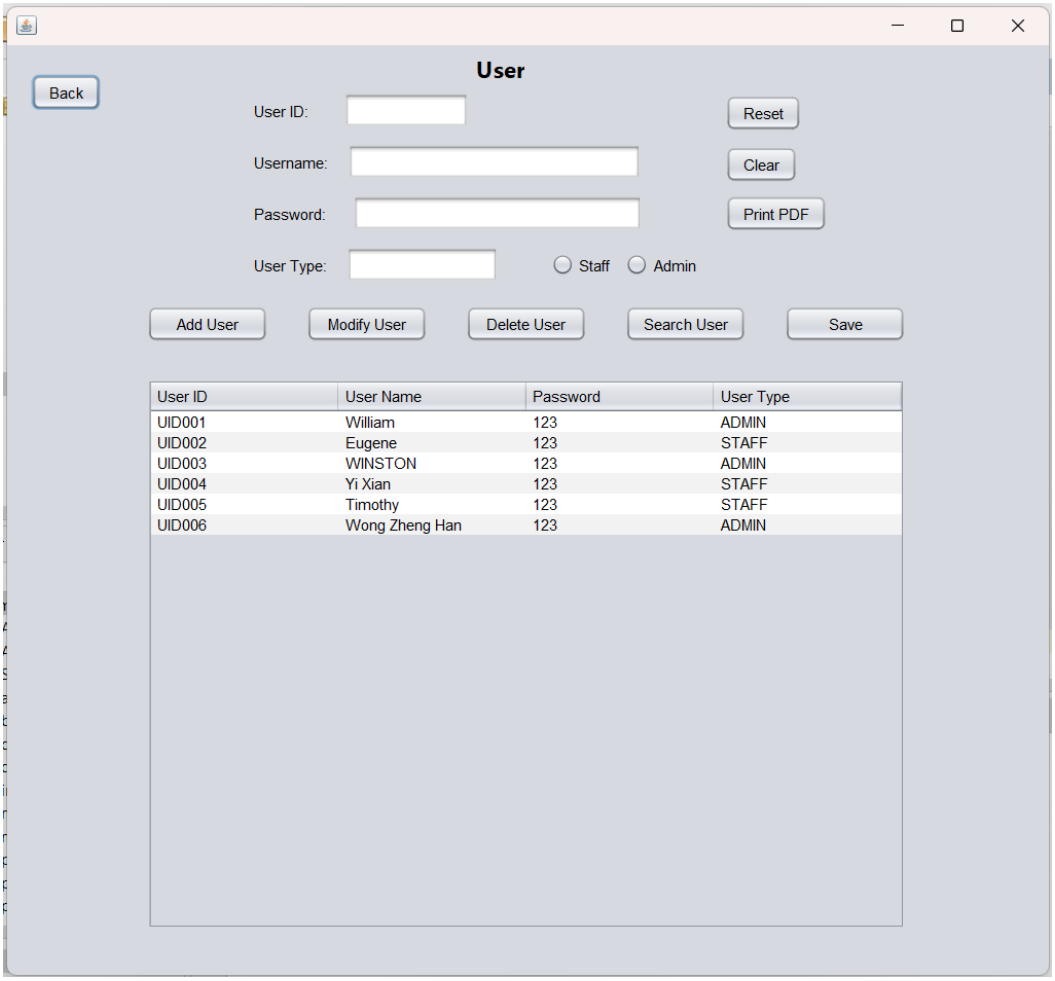



Diagram 4: User

This is the User JFrame in the PPE Inventory Management System which admin can only access. The admin can add user, modify user detail, delete user and search user detail. To add user, admin must select Add User button. To modify user, admin must select user record then select modify user button. To delete user, admin must select user record then select delete user button. To search user detail, admin can enter detail in the input fields then select search user button. The save button writes the updated table into the users text file. Additionally, the reset button is used to reset the filtered table back to the original table. There is also a clear button to clear the input fields. Lastly, there is a print pdf button which saves the table as a pdf file for admin.

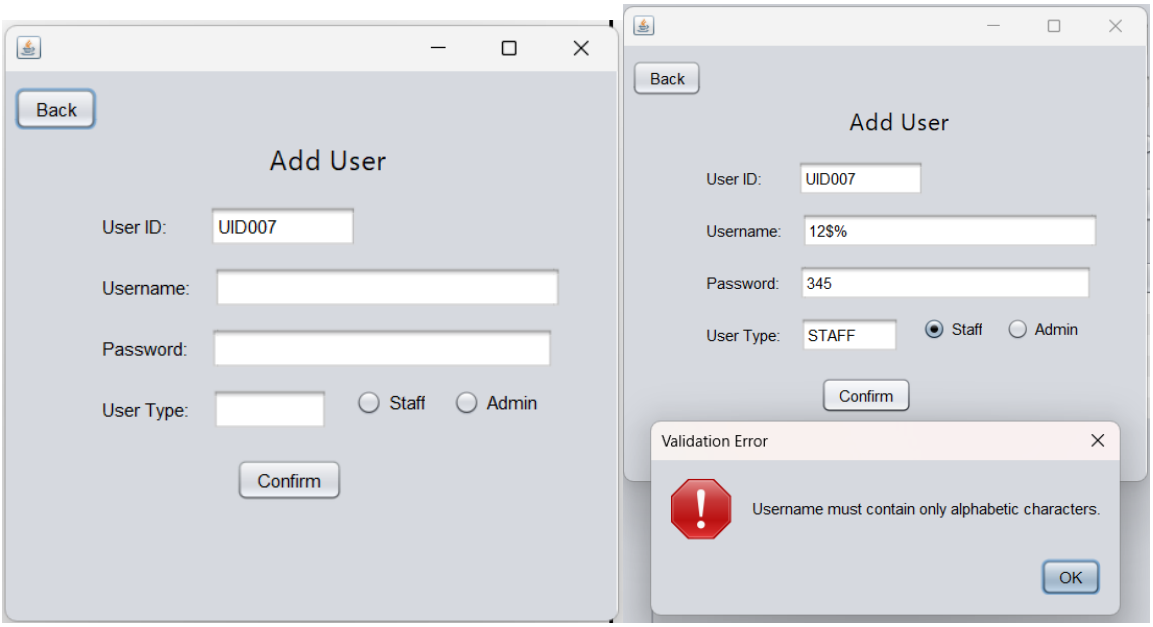



Diagram 5: Add User

This is the Add User JFrame after admin selected add user button. The user id is automatically generated but admin must enter username, password and select the user type radio button (staff or admin). Once completed, admin must select confirm and select save button in User JFrame to update users text file. The input is validated for username to only accept alphabets.

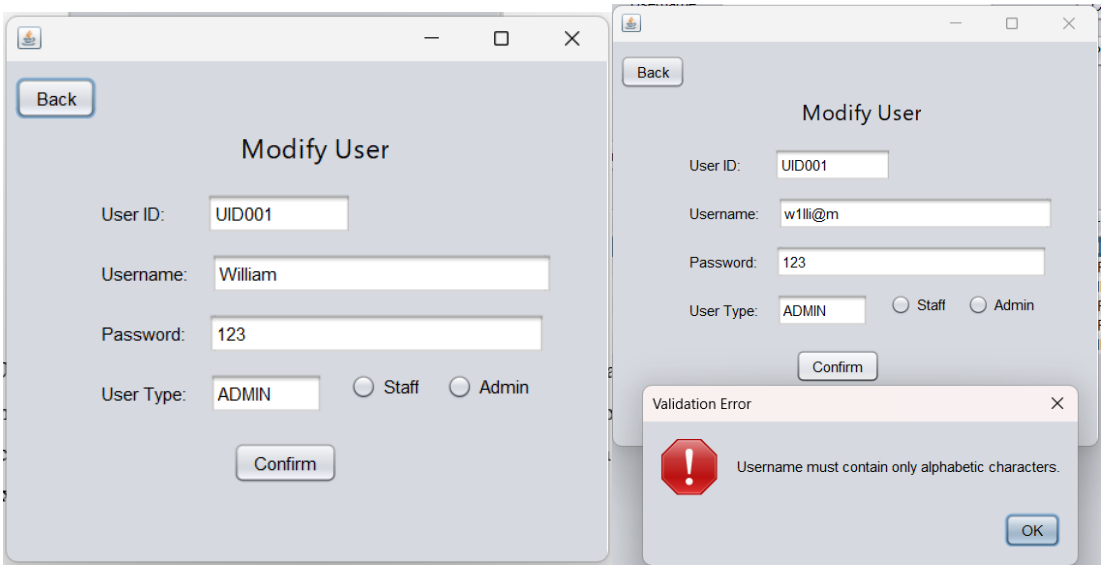



Diagram 6: Modify User

This is the Modify User JFrame after admin selected modify user button. The user id cannot be modified but admin can modify username, password and user type (staff or admin). Once

completed, admin must select confirm and select save button in User JFrame to update users text file. The input is validated for username to only accept alphabets.

PPE

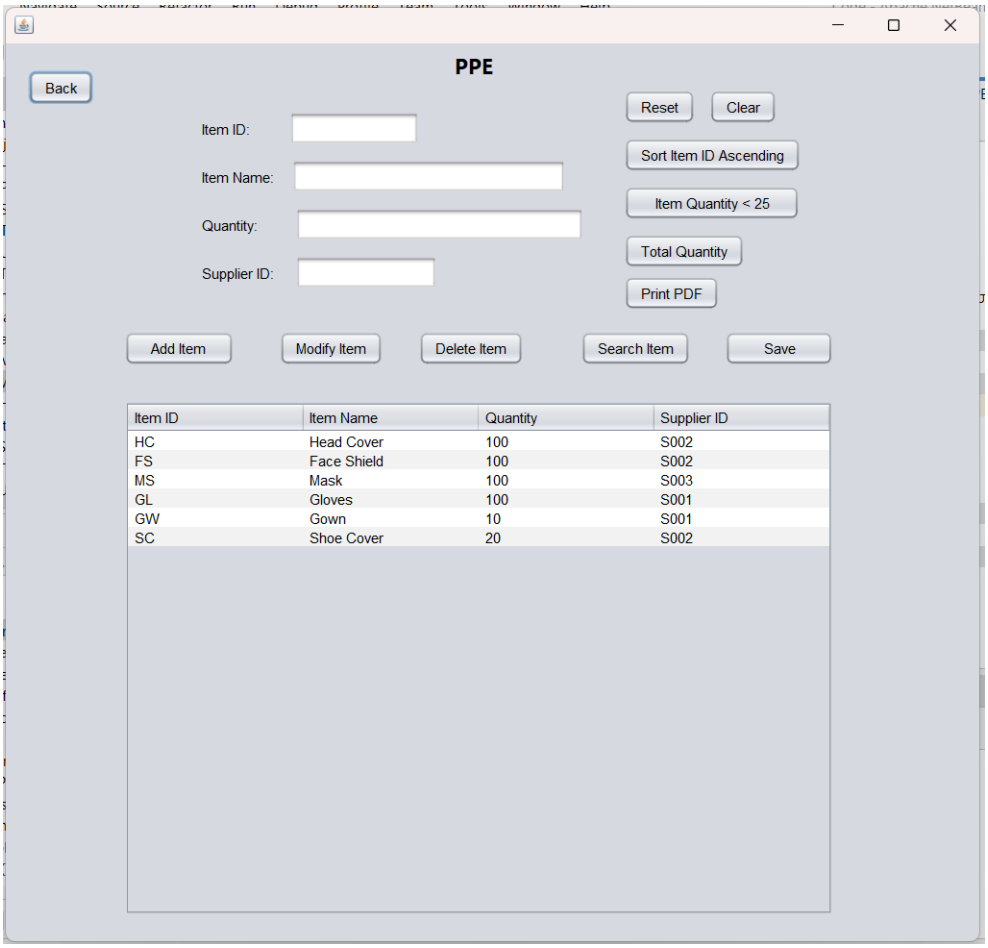



Diagram 7: PPE

This is the PPE JFrame in the PPE Inventory Management System, where user can add item, modify item detail, delete item and search item detail. To add item, user must select Add Item button. To modify item, user must select item record then select modify item button. To delete item, user must select item record then select delete item button. To search item detail, user can enter detail in the input fields then select search item button. The save button writes the updated table into the ppe text file. Additionally, the reset button is used to reset the filtered table back to the original table. There is also a clear button to clear the input fields. There is a “sort item id ascending” button, “item quantity < 25” to display item with quantity less than 25 and “total quantity” button to display total quantity of all items. Lastly, there is a print pdf button which saves the table as a pdf file for user.

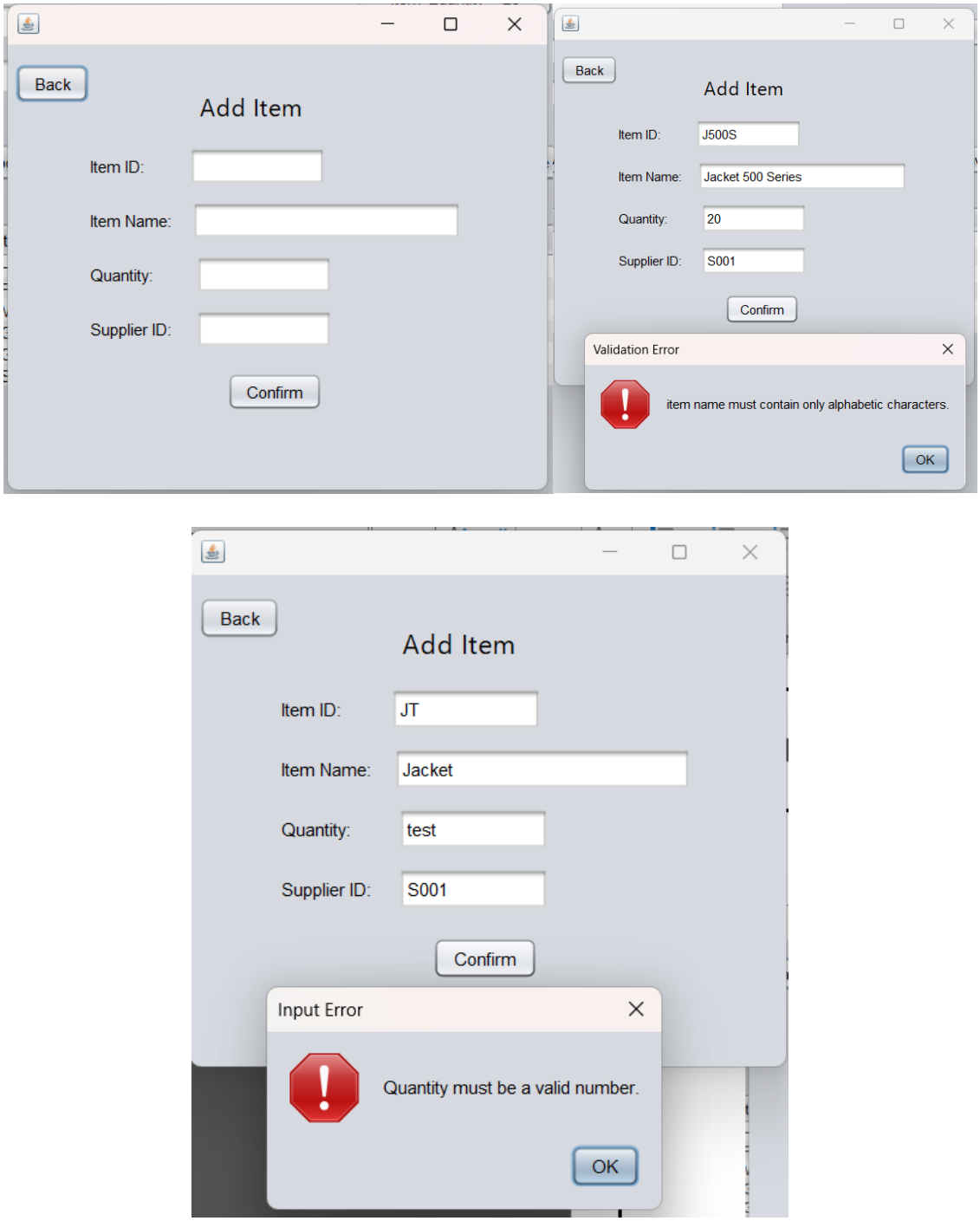



Diagram 8: Add Item

This is the Add Item JFrame after user selected add item button. The user must enter item id, item name, quantity and supplier id. Once completed, user must select confirm and select save button in PPE JFrame to update ppe text file. The input is validated for item name to only accept alphabets and quantity to accept integer.

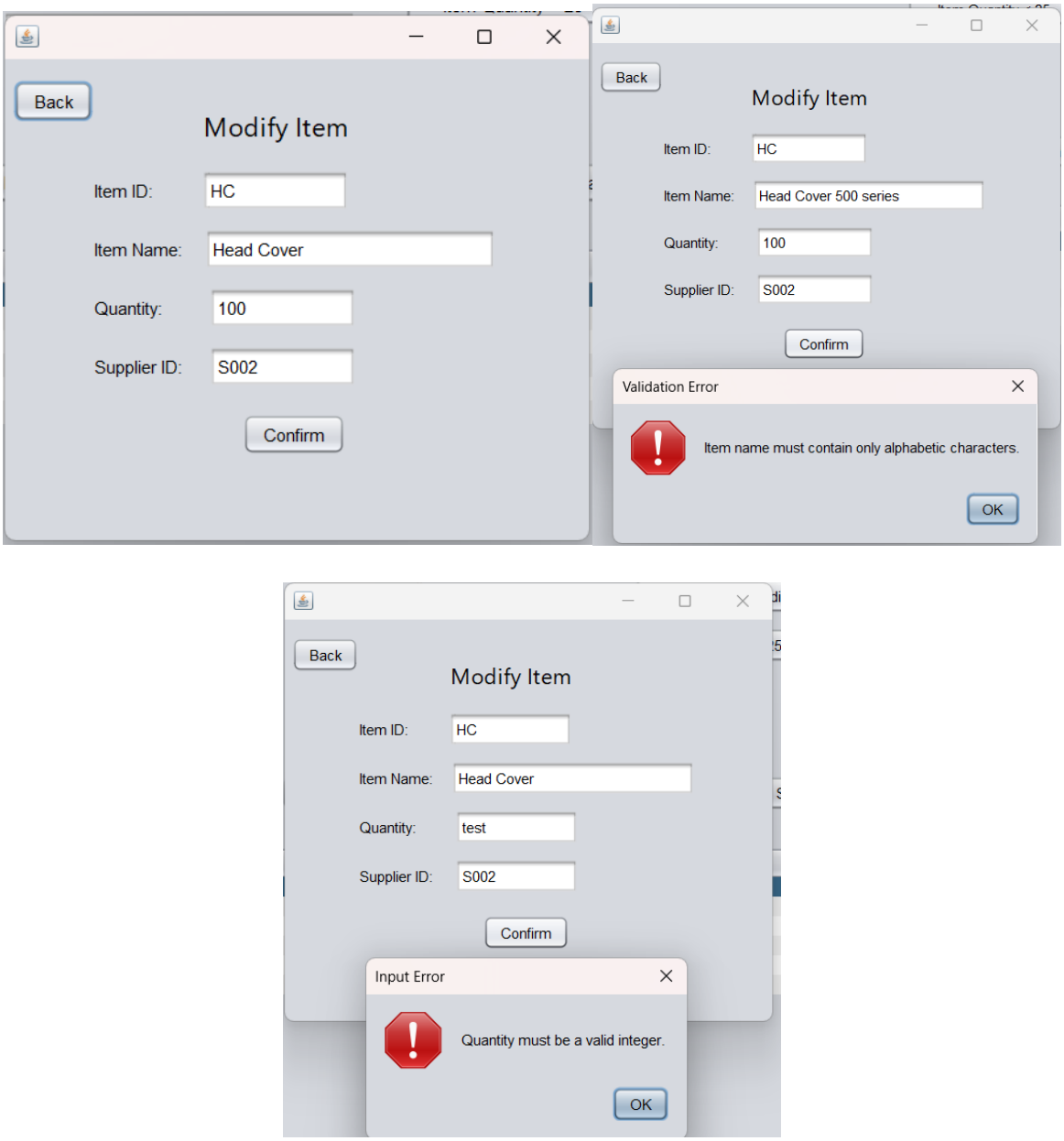



Diagram 9: Modify Item

This is the Modify Item JFrame after user selected modify item button. The user can modify item id, item name, quantity and supplier id. Once completed, user must select confirm and select save button in PPE JFrame to update ppe text file. The input is validated for item name to only accept alphabets and quantity to accept integer.

Supplier

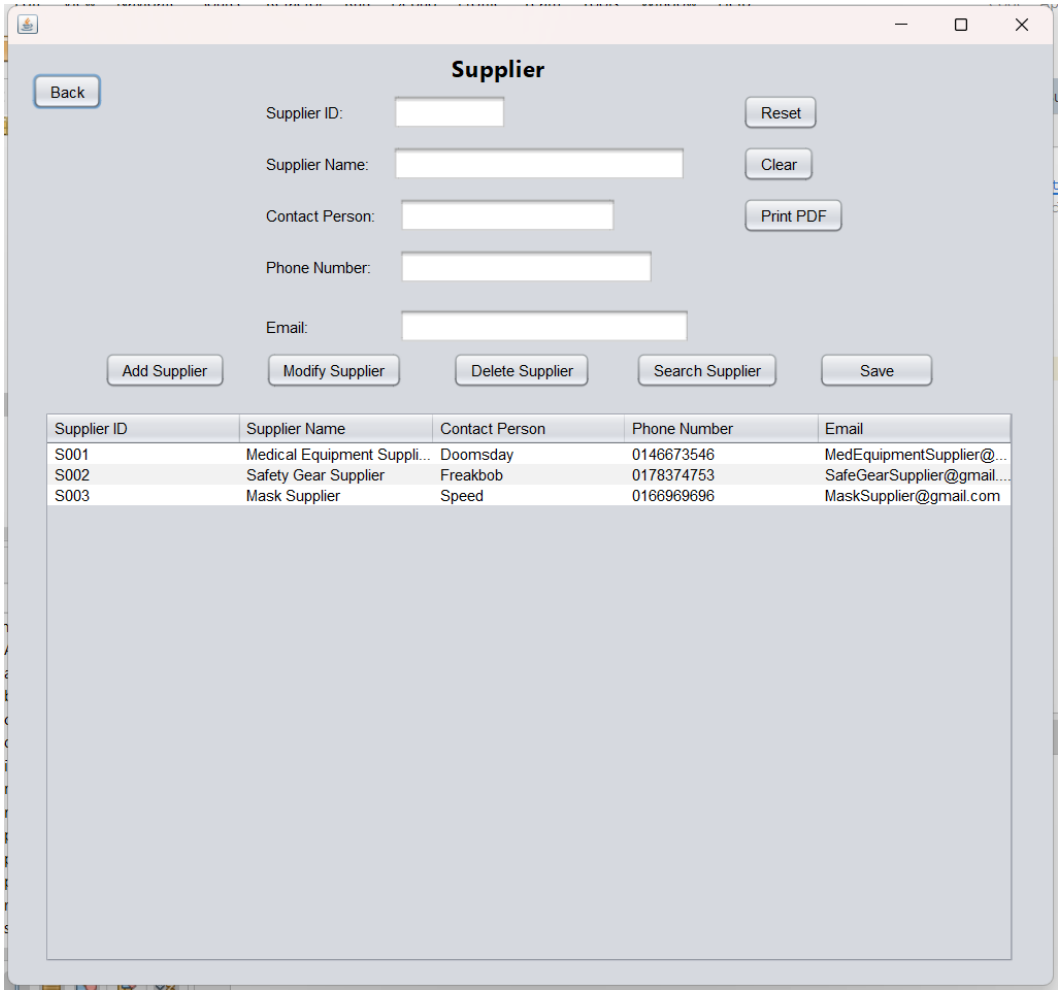



Diagram 10: Supplier

This is the Supplier JFrame in the PPE Inventory Management System, where user can add supplier, modify supplier detail, delete supplier and search supplier detail. To add supplier, user must select add supplier button. To modify supplier, user must select supplier record then select modify supplier button. To delete supplier, user must select supplier record then select delete supplier button. To search supplier detail, user can enter detail in the input fields then select search supplier button. The save button writes the updated table into the suppliers text file. Additionally, the reset button is used to reset the filtered table back to the original table. There is also a clear button to clear the input fields. Lastly, there is a print pdf button which saves the table as a pdf file for user.

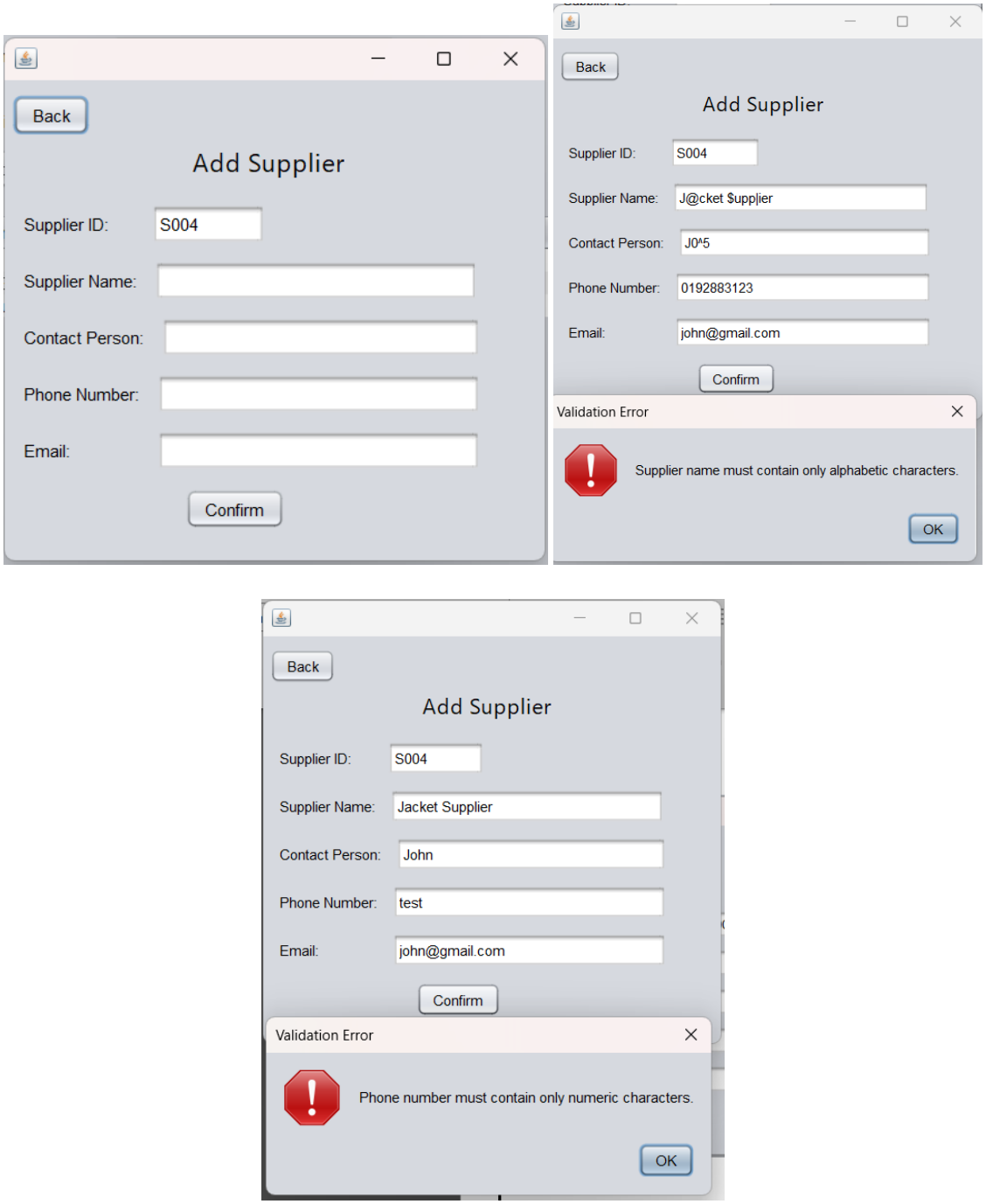



Diagram 11: Add Supplier

This is the Add Supplier JFrame after user selected add supplier button. The supplier id is automatically generated but user must enter supplier name, contact person, phone number and email. Once completed, user must select confirm and select save button in Supplier JFrame to update suppliers text file. The input is validated for supplier name and contact person to only accept alphabets, while phone number is validated to accept integer.

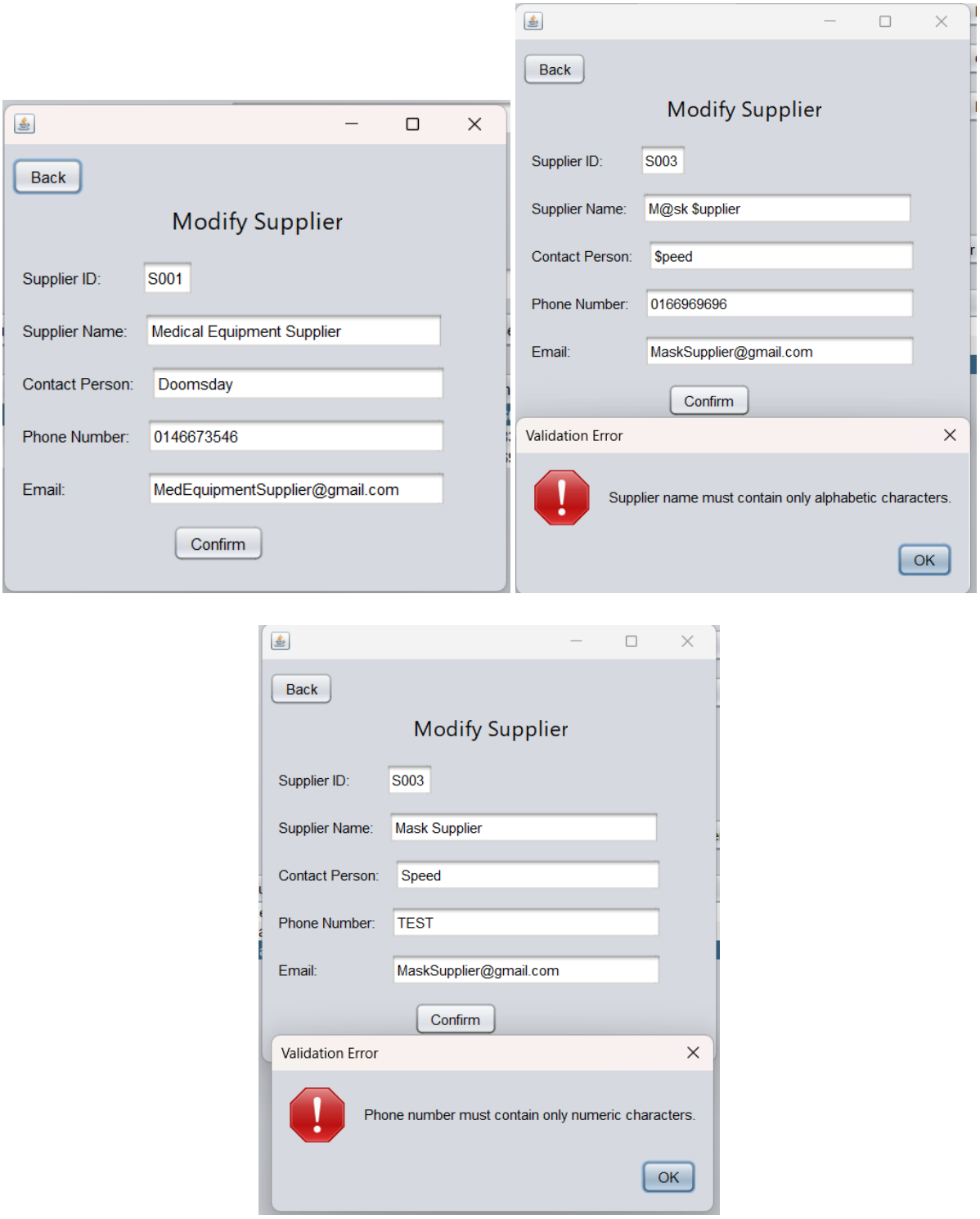



Diagram 12: Modify Supplier

This is the Modify Supplier JFrame after user selected modify supplier button. The supplier id cannot be modified but user can modify supplier name, contact person, phone number and email. Once completed, user must select confirm and select save button in Supplier JFrame to update suppliers text file. The input is validated for supplier name and contact person to only accept alphabets, while phone number is validated to accept integer.

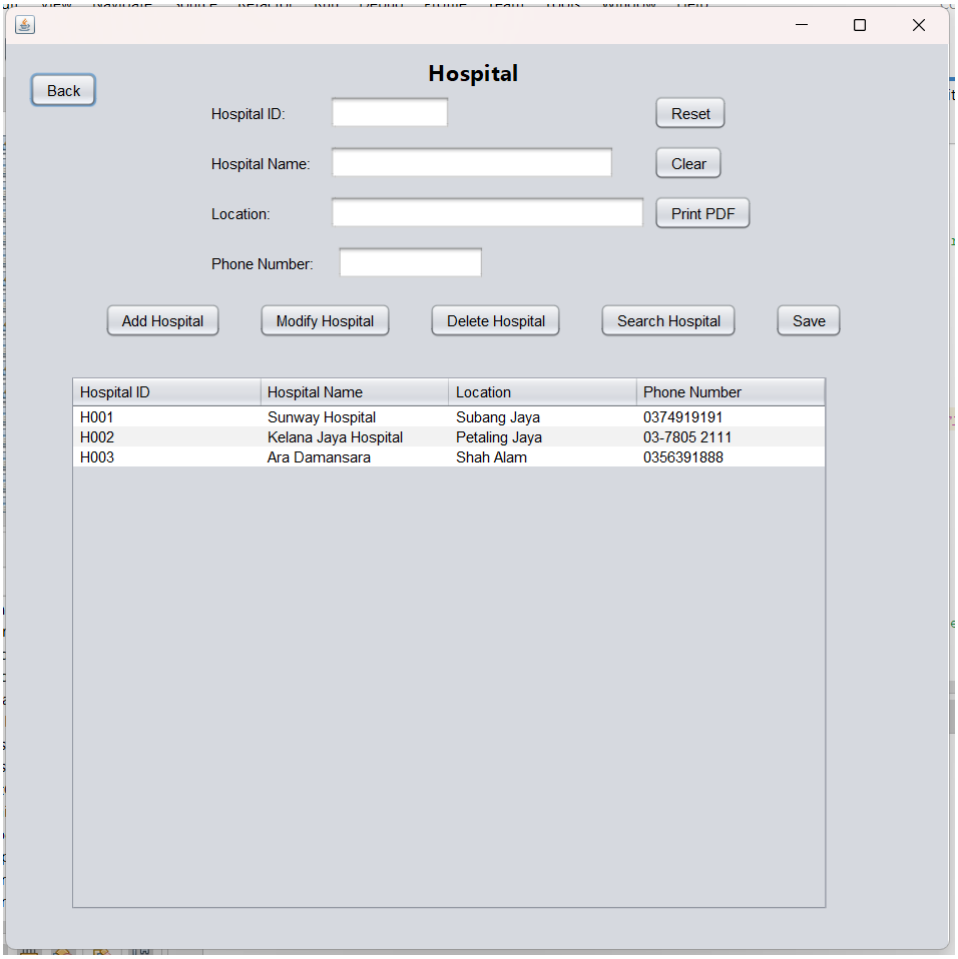



Diagram 13: Hospital

This is the Hospital JFrame in the PPE Inventory Management System, where user can add hospital, modify hospital detail, delete hospital and search hospital detail. To add hospital, user must select add hospital button. To modify hospital, user must select hospital record then select modify hospital button. To delete hospital, user must select hospital record then select delete hospital button. To search hospital detail, user can enter detail in the input fields then select search hospital button. The save button writes the updated table into the hospitals text file. Additionally, the reset button is used to reset the filtered table back to the original table. There is also a clear button to clear the input fields. Lastly, there is a print pdf button which saves the table as a pdf file for user.

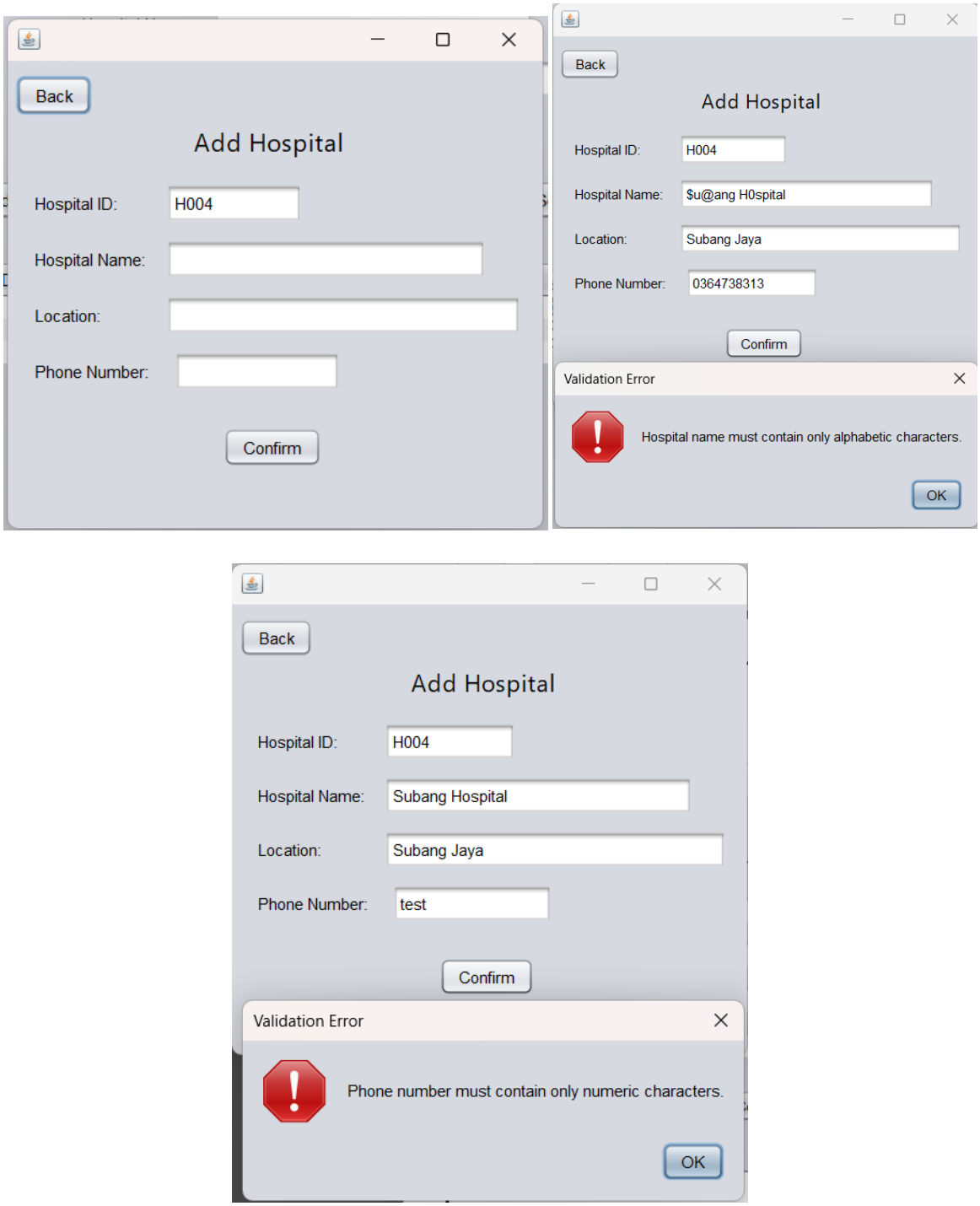



Diagram 14: Add Hospital

This is the Add Hospital JFrame after user selected add hospital button. The hospital id is automatically generated but user must enter hospital name, location and phone number. Once completed, user must select confirm and select save button in Hospital JFrame to update hospitals text file. The input is validated for hospital name to only accept alphabets and phone number to accept integer.

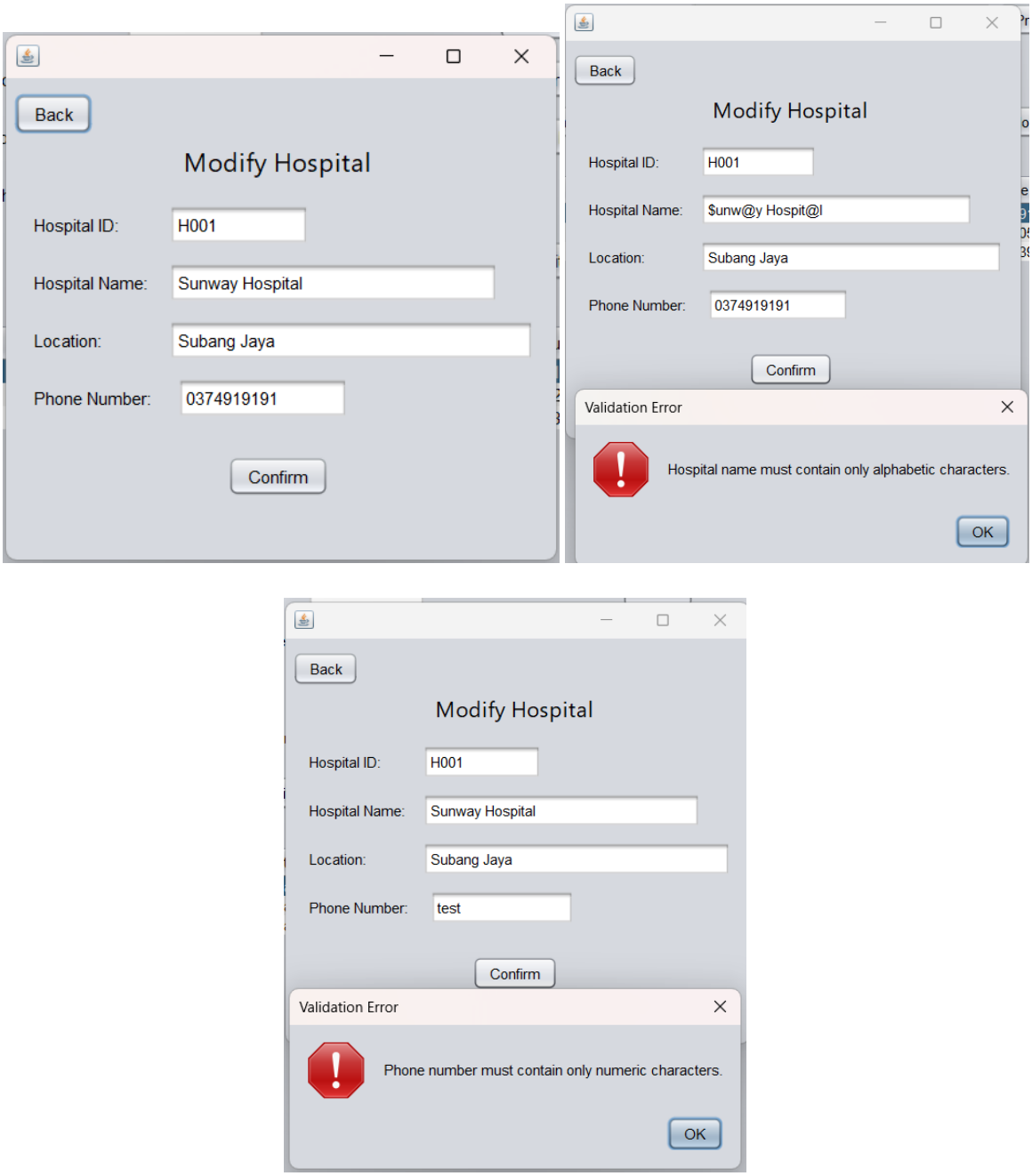
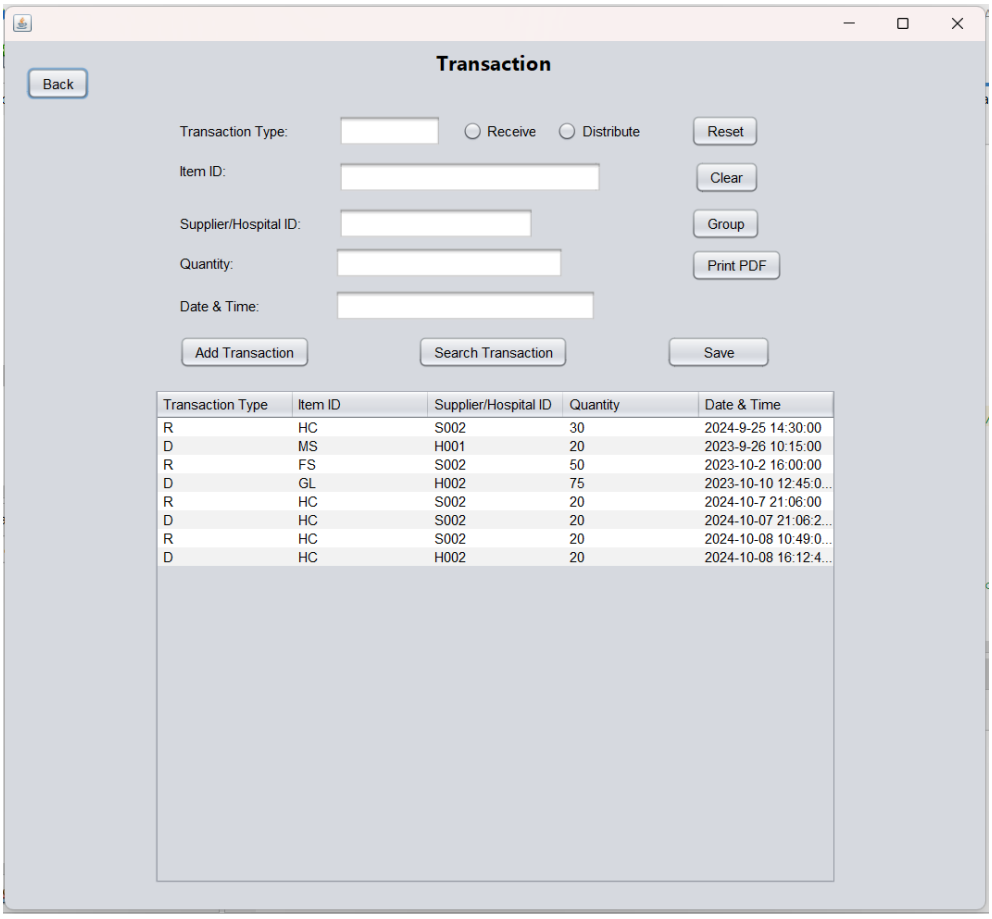



Diagram 15: Modify Hospital

This is the Modify Hospital JFrame after user selected modify hospital button. The hospital id cannot be modified but user can modify hospital name, location and phone. Once completed, user must select confirm and select save button in Hospital JFrame to update hospitals text file. The input is validated for hospital name to only accept alphabets and phone number to accept integer.



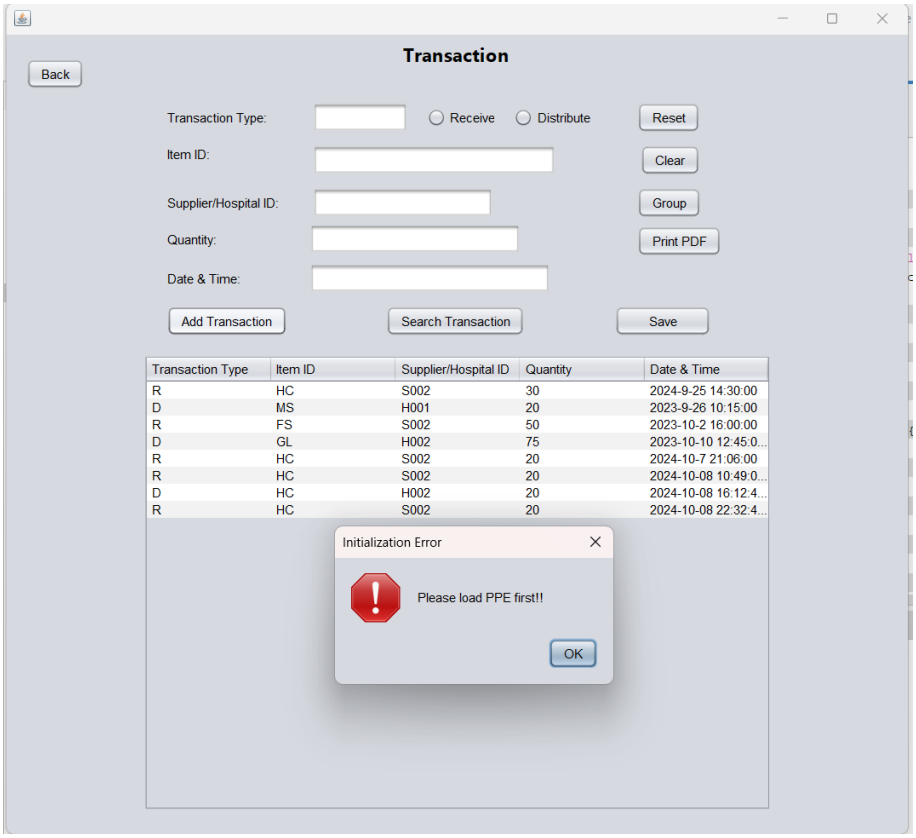




Diagram 16: Transaction

This is the Transaction JFrame in the PPE Inventory Management System, where user can add transaction and search transaction detail. To add transaction, user must select add transaction button. To search transaction detail, user can enter detail in the input fields then select search transaction button. The save button writes the updated table into the transaction text file. Additionally, the reset button is used to reset the filtered table back to the original table. There is also a clear button to clear the input fields. There is also a group button which groups transaction records for the same supplier/hospital id. Lastly, there is a print pdf button which saves the table as a pdf file for user. If user did not open ppe before adding transaction, there will be an error message displayed to user, asking user to load ppe as ppe has no content since it is not initialized.

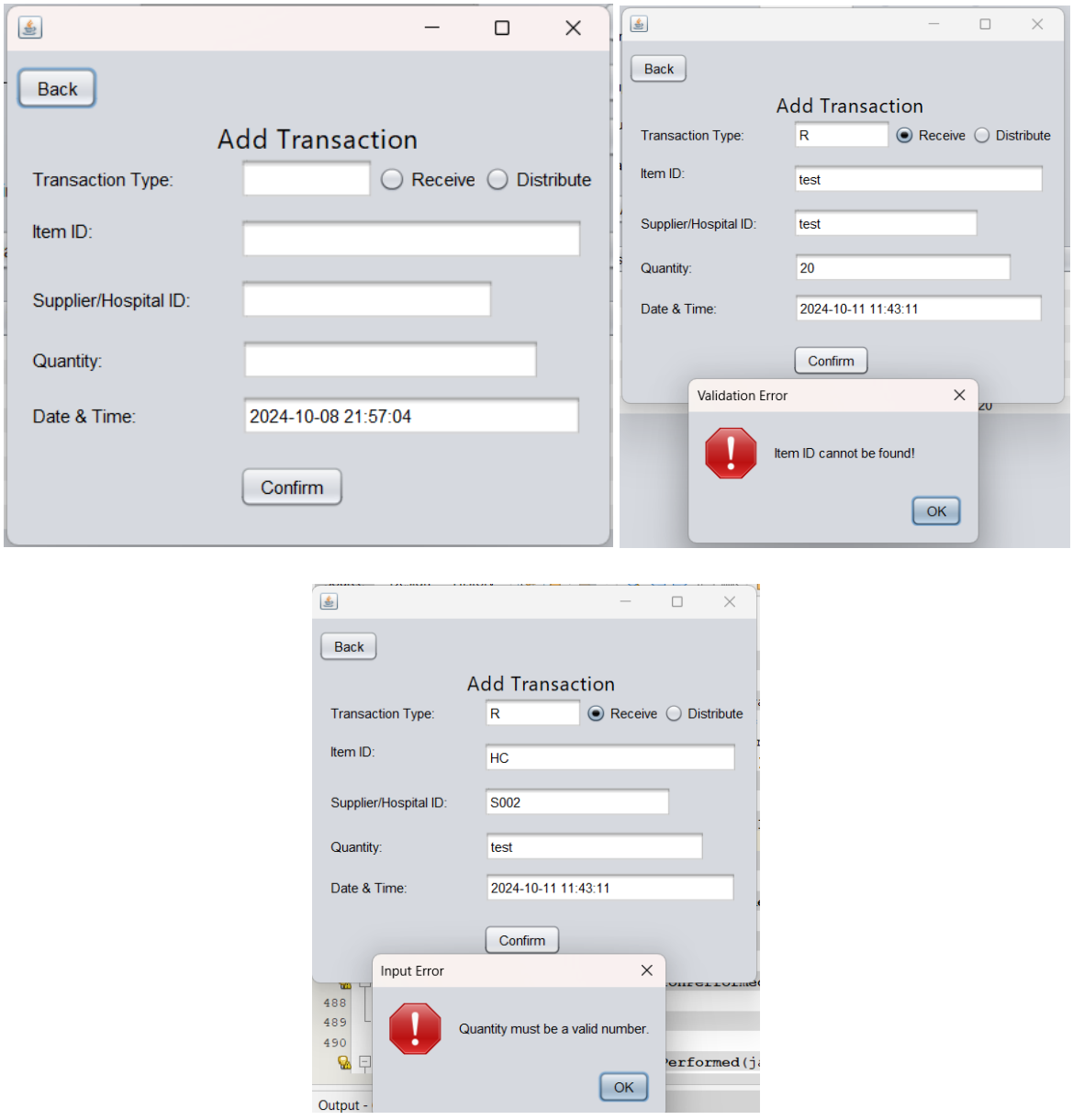



Diagram 17: Add Transaction

This is the Add Transaction JFrame after user selected add transaction button. The date is automatically generated but user must select transaction type (receive or distribute), item id, hospital id (supplier code is automatically generated according to item id entered) and quantity. Once completed, user must select confirm and select save button in Transaction JFrame to update transactions text file. After transaction is made, the table in PPE JFrame and PPE text file will be updated. However, user must first access into PPE JFrame before adding transaction. The input is validated for item id found in ppe and quantity to be integer.

# 3.0 Implementation Code Description & Justification

## 3.1 Object Oriented Programming Concepts

There are six object-oriented programming (OOP) concepts and they are class, object, encapsulation, abstraction, inheritance and polymorphism. These concepts allow us to create working variables and procedures and then reuse all or some of them again without sacrificing security. (Stackify Team, 2024)

The definitions for the 6 following OOP concepts are:

1.Class

Classes are prototypes for creating objects. The object's behaviour and structure are defined by its class. Class can be thought as an object creation recipe. Similarly to how a cookbook instructs you on which ingredients to use, how to prepare them, and how long to cook them, a class describes an object's characteristics and other useful information. Class let you make things with predictable and consistent behaviors. A class have unique properties, objects, and functions. (Avdhoot Fulsundar, 2023)

```
public class a_AddUser extends javax.swing.JFrame {

    private DefaultTableModel mainTableModel;
    private int lastUserID;

    public a_AddUser(DefaultTableModel tableModel) {
        this.mainTableModel = tableModel;
        this.lastUserID = calculateLastUserID();
        initComponents();
        setUserID();
        usertypeINP.setEditable(false);
        useridINP.setEditable(false);


    }

    private int calculateLastUserID() {
        int lastID = 0;
        for (int i = 0; i < mainTableModel.getRowCount(); i++) {
            String userID = mainTableModel.getValueAt(i, 0).toString();
            if (userID.startsWith("UID")) {
                int currentID = Integer.parseInt(userID.substring(3));
                if (currentID > lastID) {
                    lastID = currentID;
                }
            }
        }
        return lastID;
    }

    private void setUserID() {
        String newUserId = "UID00" + (lastUserID + 1);
        useridINP.setText(newUserId);
    }
```

Diagram 18: Add User JFrame

This is an example of class concept where in this JFrame which act as an class has few objects with functionalities.

2.Object

Object is an instance of a class where it contains data and method to work on the data. When a class is created there is no memory allocated. After an object is created, then the memory will be allocated. An object consists of three things a variable name, data and behaviour that describes the object. (Ankid Nishad, 2024) Objects can interact with data and code without knowing the details, it is sufficient to understand what message it received and type of response returned. (Sambhav228, 2023)

```
private void resetBTNActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) userTable.getModel();
    TableRowSorter<DefaultTableModel> sorter = new TableRowSorter<>(model);
    userTable.setRowSorter(sorter);

    sorter.setRowFilter(null);

    useridINP.setText("");
    usernameINP.setText("");
    userpassINP.setText("");
    usertypeINP.setText("");
}
```

Diagram 19: User Table Creation

This is an example of object concept where an object user table is created.

3.Abstraction

Abstraction breaks down large systems into small and manageable parts, allowing programmers to design complicated systems. A programmer can build many kinds of objects, such as variables, functions, and data structures. In order to define the objects, programmers can also design many classes of objects. For example, an address could be a class of variable. Every address object may have a name, street, city, and zip code specified by the class. In this instance, the objects may be supplier addresses, customer addresses, or hospital addresses. Furthermore, abstraction offers a way to hide a class's or method's implementation specifics from the public and give clients a more user-friendly interface. (Stackify Team, 2024)

```
private void userBTNActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    AdminUser adminU = new AdminUser();
    adminU.setVisible(true);
}

private void ppeBTNActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    AdminPPE adminPPE = new AdminPPE();
    adminPPE.setVisible(true);
}

private void suppliersBTNActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    AdminSupplier adminS = new AdminSupplier();
    adminS.setVisible(true);
}

private void transactionsBTNActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    AdminPPE adminPPE = new AdminPPE();
    AdminTransaction adminT = new AdminTransaction();
    adminT.setAdminPPEInstance(adminPPE);
    adminT.setVisible(true);
}

private void hospitalBTNActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    AdminHospital adminH = new AdminHospital();
    adminH.setVisible(true);
}
```

Diagram 20: Admin Homepage

This is an example of abstract concept where a complicated system breaks down into five different components with different functionalities.

4.Encapsulation

Encapsulation keeps field within a class private so that the public can't access to these fields. It is a secure method that stores the data and code to be safe within its own class. Programmers can reuse variables and data without allowing public access to the data. Original code can also be modified without affecting others code. It provides 3 main advantages like data hiding, modularity and flexibility. It hides implementation details to protect data from unauthorized access. Modularity where it can break down complex system into smaller parts making the code easier to manage. It provides a controlled interface for connectivity with class without affecting the external interface. (Stackify Team, 2024)

```
private void populateTransTable() {
    try (BufferedReader br = new BufferedReader(new FileReader("transaction.txt"))) {
        String record;
        while ((record = br.readLine()) != null) {
            String[] hospital = record.split(",");
            transTableModel.addRow(hospital);
        }
    } catch (FileNotFoundException e) {
        JOptionPane.showMessageDialog(this, "File doesn't exist!", "Error", JOptionPane.ERROR_MESSAGE);
    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error occurred while reading the file!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

Diagram 21: Populate Table Code

This is an example of encapsulation for writing to text file without showing it to the external interface.

5.Inheritance

New classes can inherit attributes of another existing class with inheritance concept. This concept helps in shortening the process of remaking class where it can use inheritance concept on previous work. Advantages of inheritance are the reusability, polymorphism and flexibility. Programmers can create a subclass to reuse the code and functionality of the main class where it can help save a lot of time. Polymorphism can also can be use inheritance concept to make generic coding easier. It flexible enough to let programmers add new feature to an existing class without alternating the existing code. (Stackify Team, 2024)

```
public class a_AddUser extends javax.swing.JFrame {

    private DefaultTableModel mainTableModel;
    private int lastUserID;

    public a_AddUser(DefaultTableModel tableModel) {
        this.mainTableModel = tableModel;
        this.lastUserID = calculateLastUserID();
        initComponents();
        setUserID();
        usertypeINP.setEditable(false);
        useridINP.setEditable(false);


    }
```

Diagram 22: User Table Connection

This is an example of inheritance concept where the Add User JFrame inherit the table model used by User JFrame to read the data.

6.Polymorphism

Polymorphism let users to do different method with different contexts by using the same word in Java. The example of polymorphism is method overloading and overriding. Method overloading is a method that is able to perform different functions depending on the context of the name. A different method my be performed in different ways with arguments passed to it. Method overriding allows programmers to perform different method in different ways using the same method. Benefits of polymorphism are flexibility, code reuse, and simplification. Polymorphism is able to treat different class as if they are the same class which is very flexible. Code can reuse by inheriting functionality from other classes to share the same properties. It also enables the uses of generic code that can handle different objects. (Stackify Team, 2024)

```
private void saveBTNActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) transactionTable.getModel();

    if (model.getRowCount() == 0){
        JOptionPane.showMessageDialog(this, "There are no records to save!", "Warning", JOptionPane.WARNING_MESSAGE);
    }

    try (BufferedWriter bw = new BufferedWriter(new FileWriter("transaction.txt"))){
        for (int i = 0; i < model.getRowCount();i++){
            String transaction = model.getValueAt(i,0).toString() + "," +
            model.getValueAt(i,1).toString() + "," +
            model.getValueAt(i,2).toString() + "," +
            model.getValueAt(i,3).toString() + "," +
            model.getValueAt(i,4).toString();
            bw.write(transaction);
            bw.newLine();
        }
        bw.close();
        JOptionPane.showMessageDialog(this, "Data saved successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
    }catch(IOException ex){
        JOptionPane.showMessageDialog(this, "Error Occured while writing to File!", "Try Again!", JOptionPane.ERROR_MESSAGE);
    }
}

private void quantityINPActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void resetBTNActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) transactionTable.getModel();
    TableRowSorter<DefaultTableModel> sorter = new TableRowSorter<>(model);
    transactionTable.setRowSorter(sorter);

    sorter.setRowFilter(null);

    t_typeINP.setText("");
    itemidINP.setText("");
    suphospidINP.setText("");
    quantityINP.setText("");
    datetimeINP.setText("");
}
```

Diagram 23: Save Button and Reset Button

This example uses the polymorphism concept where save button and reset button is widely use in others JFrame.

## 4.0 Conclusion

The system we designed uses all six of the concepts of OOP to make sure that the system is user friendly and not too overwhelming for the user. Class and object concept are used for the creation of admin, staff and table functionalities. Abstract and encapsulation concept are used for functionalities in all of the JFrame to make sure that the data can only be accessed by admin and staff. Inheritance and polymorphism concept are used for the repetition of functions that are created in the JFrames.

Overall, the limitation of our project is the knowledge of programming, time for the project and experience in creating a system. We are inexperienced in creating a system and we are learning as we coding. We hope that this will be a valuable experience for our future projects.

## 5.0 References


Stackify Team (September 1, 2024). OOP Concepts in Java: Defined and Explained with Examples

https://stackify.com/oops-concepts-in-java/#:~:text=The%20main%20ideas%20behind%20Java%27s,of%20them%20without%20compromising%20security.

Avdhoot Fulsundar (May 2, 2023). Object Oriented Programming Basics – OOP, Classes, and Objects in Java

https://www.freecodecamp.org/news/object-oriented-programming-basics-oop-classes-and-objects-in-java/

Ankit Nishad (2024). Classes and Objects in OOP

https://www.enjoyalgorithms.com/blog/classe-and-object-in-oops

Sambhav228 (February 9, 2023) Introduction of Object Oriented Programming

https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/

Arraylist:

W3Schools.com. (n.d.). https://www.w3schools.com/java/java_arraylist.asp

BufferedReader:

GeeksforGeeks. (2022, May 3). *Java.io.BufferedReader class in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/

BufferedWriter:
GeeksforGeeks. (2022, August 16). *Java.io.BufferedWriter class methods in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/io-bufferedwriter-class-methods-java/

*W3Schools.com*. (n.d.-b). https://www.w3schools.com/java/java_encapsulation.asp

Comparator:

GeeksforGeeks. (2024, October 4). *Comparator Interface in Java with Examples*. GeeksforGeeks. https://www.geeksforgeeks.org/comparator-interface-java/

Data Validation:

*Jmix.io*. (n.d.). jmix.io. https://www.jmix.io/cuba-blog/validation-in-java-applications/

Date & Time:

*How to get current date and time in Java - Javatpoint*. (n.d.). www.javatpoint.com. https://www.javatpoint.com/java-get-current-date

FileReader:

*Java FileReader (With Examples)*. (n.d.). https://www.programiz.com/java-programming/filereader

FileWriter:
GeeksforGeeks. (2022, February 10). *FileWriter Class in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/filewriter-class-in-java/

Focus Listener:

*Is this Correct way to add FocusListener to JTextFields in java?* (n.d.). Stack Overflow. https://stackoverflow.com/questions/26423468/is-this-correct-way-to-add-focuslistener-to-jtextfields-in-java

Hashmap:

*W3Schools.com*. (n.d.-c). https://www.w3schools.com/java/java_hashmap.asp

Printer Job:

*PrinterJob (Java Platform SE 8 )*. (2024, June 5).
https://docs.oracle.com/javase/8/docs/api/java/awt/print/PrinterJob.html

*java PrinterJob not printing to fit paper*. (n.d.). Stack Overflow.
https://stackoverflow.com/questions/27029166/java-printerjob-not-printing-to-fit-paper

Table:

*How to use tables (The JavaTM Tutorials > Creating a GUI with swing > Using swing components)*. (n.d.).
https://docs.oracle.com/javase/tutorial/uiswing/components/table.html#sorting

*RowFilter.regexFilter multiple columns*. (n.d.). Stack Overflow.
https://stackoverflow.com/questions/2494868/rowfilter-regexfilter-multiple-columns

Youtube:

BoostMyTool. (2021, November 28). *Create Java Application with JTable and Form using Swing GUI Builder of Netbeans IDE (+ Source Code)* [Video]. YouTube.
https://www.youtube.com/watch?v=ohEFX1OngxE

## 6.0 Workload Matrix

| Member | Task |
|---|---|
| Wong Kap Onn (TP074292) | • Introduction<br>• Design Description<br>• Member 1 Task<br>• Admin Task |
| Tan Yi Han (TP070378) | • Implementation Code Justification<br>• Conclusion<br>• Member 2 Task<br>• Staff Task |