A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

GROUP ASSIGNMENT

CT038-3-2 OODJ

JAVA PROGRAMMING

UC2F2008CS

Name	TP No
Hong Jun Khuan	TP050852
Soh De Hui	TP049307

Contents

Introduction

Programming languages has become a hot topic as it is an important skill for all people, whether or not they are IT specialists. A lot of programming languages have been published over time. Java is one of the most powerful programming languages in all time.

Java is a language and framework for programming that Sun Microsyste published in 1995. If you have installed Java on several applications and websites would not be able to function in the hardware or software environment where your program runs. In addition, Java is known as a fast, safe and object-oriented programming language commonly used in Desktop Applications, Web Applications, Enterprise Applications, Smartphone, Embedded System, Smart Card, Robotics and Games, and 3 billion devices are currently running by using Java according to Java introducer.

Java application is also known as a desktop application or window-based application. Speedzz courier is a courier service application that is designed as a desktop application which helps to make the courier service more efficiency. This assignment is basically for a company called Speedzz which tend to create a parcel courier system which allows them to improve their current system. As they have required a feature which is every customer would have allow to search for their parcel. As a result, Java application is used to apply all the OOP techniques to achieve the functionalities:

1. Login access
2. User Login
3. Parcel Entry
4. Assign Delivery staff
5. Feedback entry
6. View Status
7. View Report
8. Create orders
9. Search Orders
10. Rating Entry

System Design

Use case diagram

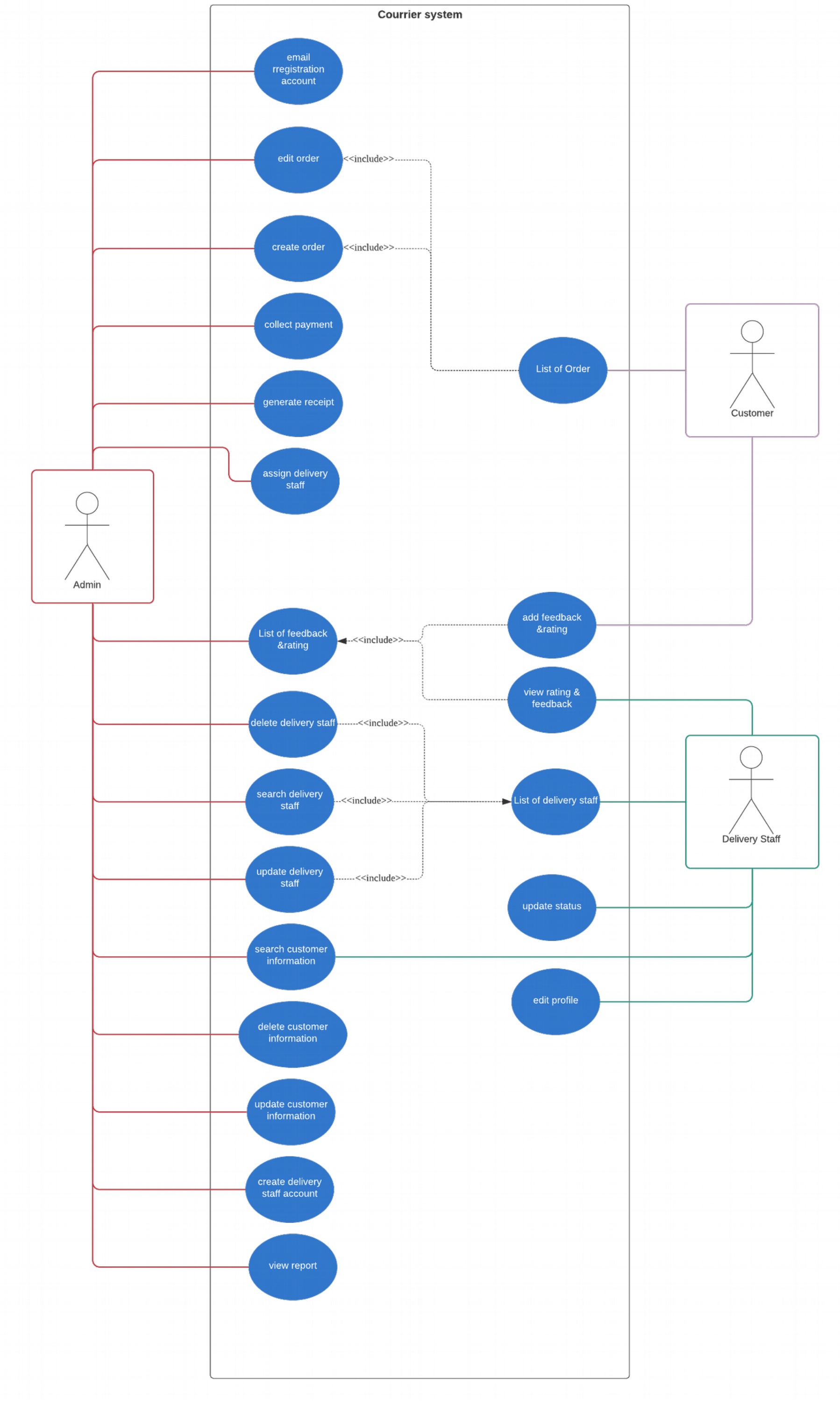


Figure 1 use case diagram

Use case specification

Use case	Brief Description	Actors	Preconditions	Main Flow	Alternative Flows
Email register account	This use case allows admin to send email to customer to inform their account information	admin	Admin provides customer name and password for customer to login	-The use case begins when visitor creates an order. -The admin will add send the email by adding customer information to create account.	The customer email is given wrongly, and the email could not send.
Edit order	This use case allows admin to edit the order information that have been made	admin	Admins edit the order that have been made	-Customer will have to approach to admin to change the details -Admin will help customer to make changes based on the customer requirement	The order details which cannot be edited like the orderID
Create order	This use case is used to create order by admin for user	admin	Admin creates order when customer approach for delivery	-Customer makes order with admin by filling in the order details	-customer does not have account -order details is not filled in properly
Collect payment	This use case is used to calculate the price of the delivery by weight.	Admin	Admin will measure the item weight and choose the range of weight with fixed price	-Customer makes payment when the order is makes and weight is measured to proceed the delivery	-Customer is not well prepared to pay - first time customer
Generate receipt	This use case is used to generate receipt for customer	Admin	Admin will print the receipt when the order is successfully assigned	-Customer makes payment and complete the order -Admins print the receipt from the order interface	-Customer does not want receipt
Assign delivery staff	This use case is used to assign delivery staff for order delivery	Admin	Admin will assign the order to delivery staff after the order create successfully	-Admins assign the order to delivery staff while the order will be removed from the admin interface and shown on the delivery staff	-Delivery staff is all on full slot.
Add feedback & rating	This use case is used for customer to write their feedback based on their orders that have been made after the order is complete	Customer	Customer will fill in their feedback and rating in their own interface after the status complete	-Customers check their order status then give feedback and rating for the delivery staff	-Feedback is not related to the order
View feedback & rating	This use case is used for admin and delivery staff to view on the feedback that is given from the	Admin, Delivery Staff	Admin and delivery staff will view the feedback and rating that is filled by the customer	-Admin and delivery staff have an interface which allows them to view on the feedback that is given by the customer	-Feedback is not given by customer
Delete delivery	This use case is used for	Admin	Admin will remove	-Admin will search the	-The staff is

staff	admin to delete the delivery staff information		the delivery staff by searching their names	delivery staff name and delete the staff name that is resigned.	already resigned.
Search delivery staff	This use case is used for admin to search the specific delivery staff information	Admin	Admin will search for the delivery staff information if is requested	-requested by customer to have the delivery staff information, admin will search for the information and get to the customer	-The staff is not handling for the customer
Update delivery staffs	This use case is used for updating the delivery staff’s information	Admin	Admin will update their information when is requested by the user	-The delivery staff requested to change its own information details.	-The requested information is not allowed to change
Search customer information	This use case is used for admin to search the specific customer information	admin	Admin will search for the customer information if is requested	-requested by customer to have the customer information, admin will search for the information and change the customer	- The customer information is not in the database yet
Delete customer information	This use case is used for admin to delete the customer information	Admin	Admin will remove the customer by searching their names	-Admin will search the customer name and delete the staff name that is resigned.	- The customer information is not in the database yet
Update customer information	This use case is used for updating the customer information	Admin	Admin will update their information when is requested by the user	-The customer requested to change its own information details.	-The requested information is not allowed to change
Create delivery staff account	This use case is used for creating delivery staff account	Admin	Admin will create delivery staff account with information given.	-The admin helps the delivery staff to register their account	-The email given is incorrect
View report	The use case is used for admin to view the report for overall of delivery staff	Admin	Admin will view and generate report for overview for the delivery.	-The admin can view the report that is created by the admin to view on the overview for the month	-There are no delivery order on that month
Update status	This use case is used for updating the customer information	Delivery staff	Delivery staff will update the order status when the status is arrived	-The item has successfully sent and update the status to completed to let the user to give feedback	-The requested order is not assigned yet
Edit profile	This use case is allowing the delivery staff to edit their own profile	Delivery staff	Delivery staff can make changes on their account.	-The delivery staff would like to update their own email address to their new email address	- The email address is used by other user.

Class diagram

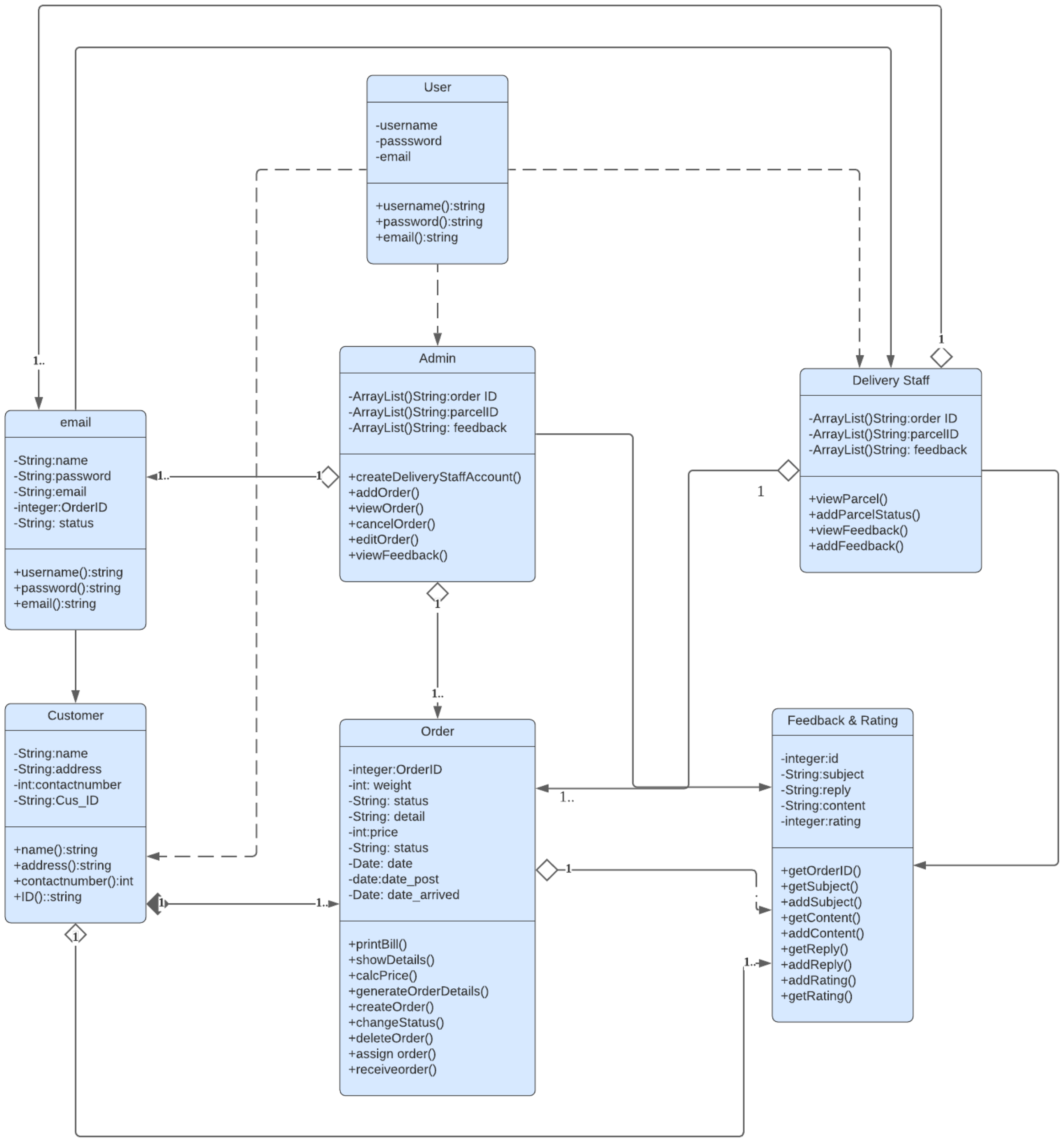


Figure 2 class diagram

System implementation

Instead of functions and logic, object-oriented programming (OOP) is a computer programming model that organizes software design around data or artefacts. An object can be defined as a data field that has unique behavior and attributes. Instead of the reasoning needed to exploit them, OOP focuses on the thing's developers want to manipulate. For program that are big, complex, and actively modified or managed, this approach to programming is well-suited.

The major ideas behind Java's Object-Oriented Programming are OOP principles in Java. They are abstraction, encapsulation, polymorphism, and inheritance. Understanding them is essential to understanding how Java operates. (What Are OOP Concepts in Java? 4 Primary Concepts, 2020) Java OOP concepts essentially allow one to build working methods and variables, then re-use all or part of them without sacrificing protection.

Therefore, each class in the program that have been implemented to describe object like User, Order, Delivery Staff, etc. Main.java and FileUtils.java is used to help terminal to interact with user while the other method is used for getting information from files.

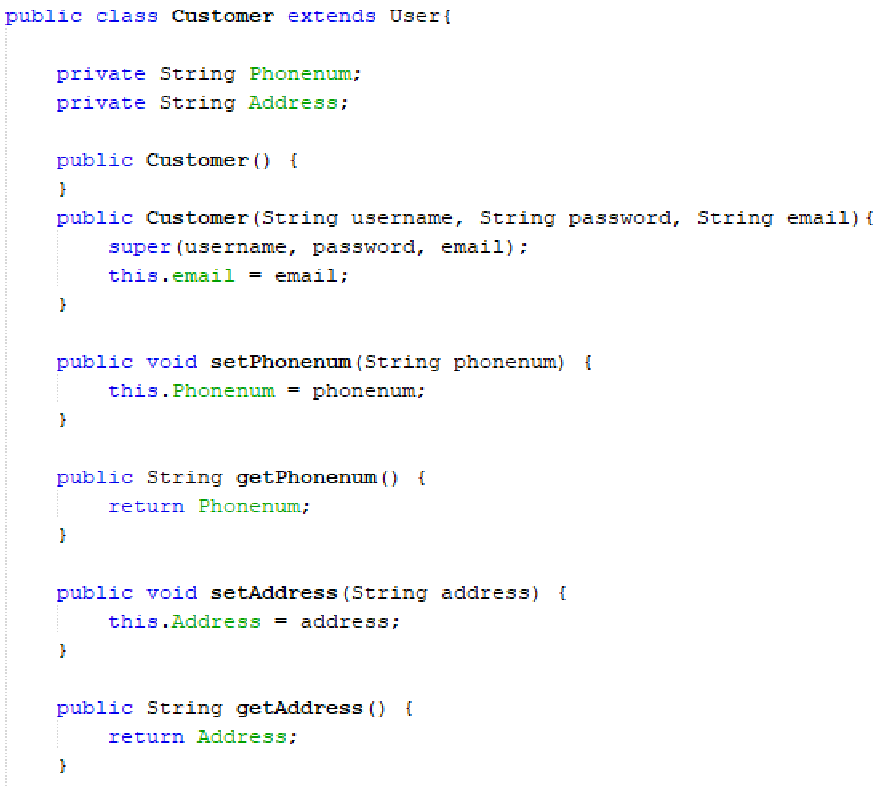
There are 4 principles of OOP which have been applied in this assignment:

1. Class and Object:

A class is an extensible program-code-template in object-oriented programming for creating objects, providing initial state values (member variables) and action implementations (member functions or methods). While an object is an abstract data type that a developer has developed. Multiple properties and methods may be included, and other objects can also be contained. Objects are known as classes in the majority of programming languages. (Classes and Objects in Java - GeeksforGeeks, 2020)

As an example, based on the project, the role of Customer is chosen which its state will be the name, ID while it comes with a behavior of sending item. Hence the object state will be stored with methods as behavior. Moreover, each class have a constructor as object is created every time with a condition of same class name because each class can have more than one constructor.

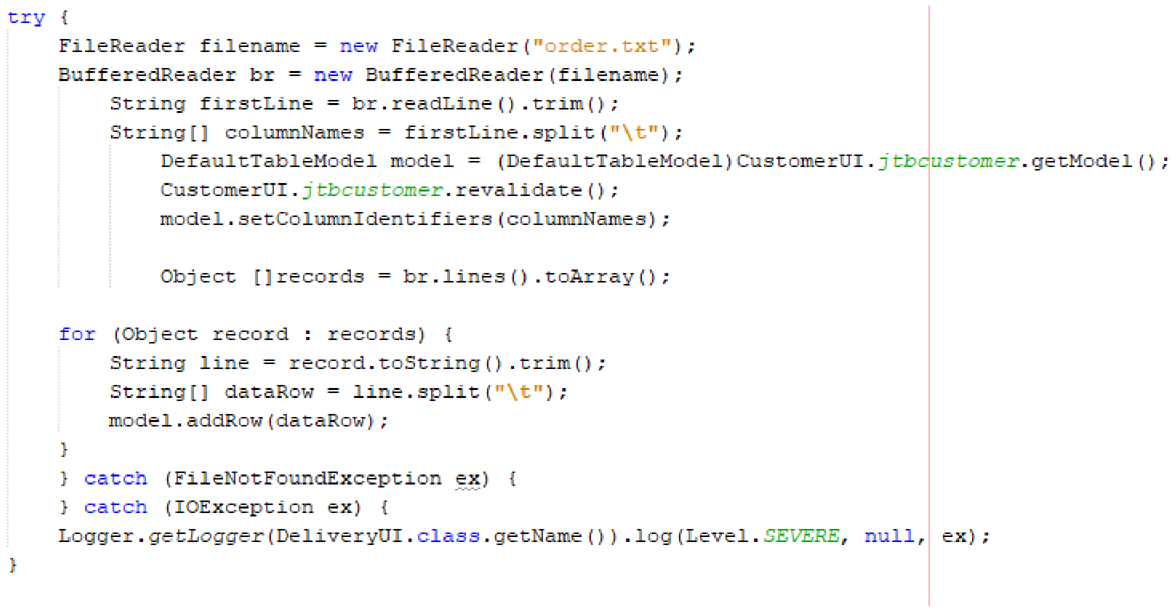
Example of class concept:



```
public class Customer extends User{

    private String Phonenum;
    private String Address;

    public Customer() {
    }
    public Customer(String username, String password, String email){
        super(username, password, email);
        this.email = email;
    }

    public void setPhonenum(String phonenum) {
        this.Phonenum = phonenum;
    }

    public String getPhonenum() {
        return Phonenum;
    }

    public void setAddress(String address) {
        this.Address = address;
    }

    public String getAddress() {
        return Address;
    }
```

Figure 3 class diagram

An example of object of customer class and data will be stored into order text file in Array List:



```
try {
    FileReader filename = new FileReader("order.txt");
    BufferedReader br = new BufferedReader(filename);
        String firstLine = br.readLine().trim();
        String[] columnNames = firstLine.split("\t");
            DefaultTableModel model = (DefaultTableModel)CustomerUI.jtbcustomer.getModel();
            CustomerUI.jtbcustomer.revalidate();
            model.setColumnIdentifiers(columnNames);

            Object []records = br.lines().toArray();

    for (Object record : records) {
        String line = record.toString().trim();
        String[] dataRow = line.split("\t");
        model.addRow(dataRow);
    }
    } catch (FileNotFoundException ex) {
    } catch (IOException ex) {
    Logger.getLogger(DeliveryUI.class.getName()).log(Level.SEVERE, null, ex);
}
```

Figure 4 store file

2. Encapsulation:

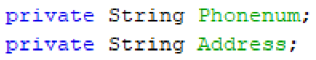
In object-oriented programming (OOP), encapsulation is one of the basic principles. It describes the concept of bundling data and methods within one unit, e.g. a class in Java, to operate on that data. This idea is also frequently used to mask an object's internal representation, or state, from the outside. (Braunschweig, 2020)This is called hiding information. This mechanism's general idea is plain. If you have an attribute that is not visible from the outside of an object, then you can mask specific information and monitor access to the internal state of the object if you bundle it with methods that provide read or write access to it.

The hiding internal information of the objects could help to protect unrelated users from changing internal data which would also reduce the system complexity. There are a few types of modifiers that can set access levels for classes, variables, methods and constructors.

- Access modifiers: public, protected, and private

- Modifier requiring override: abstract
- Modifier restricting to one instance: static

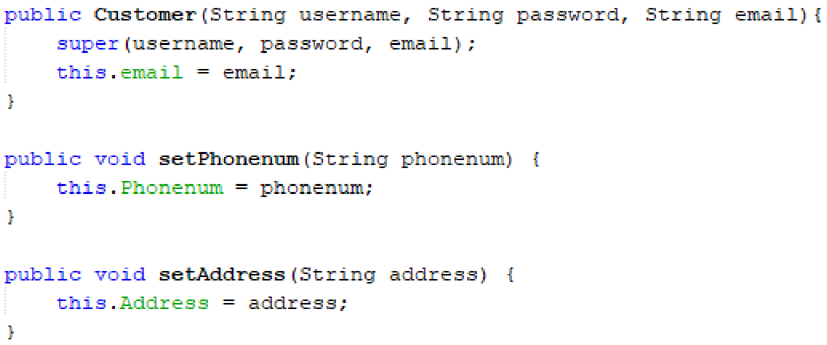
All classes of my project, that shows the objects and all variables in private, so it can be accessed with methods of class to prevent changes of object in any field of object of all classes. As an example, in customer class I have included these private fields:



```
private String Phonenum;
private String Address;
```

Figure 5 private fields in customer class

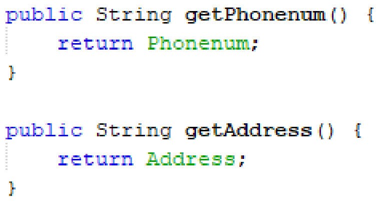
Their value can only be modified with the aid of the special method of these classes, called Mutators.



```
public Customer(String username, String password, String email){
    super(username, password, email);
    this.email = email;
}

public void setPhonenum(String phonenum) {
    this.Phonenum = phonenum;
}

public void setAddress(String address) {
    this.Address = address;
}
```

Figure 6 mutators classes

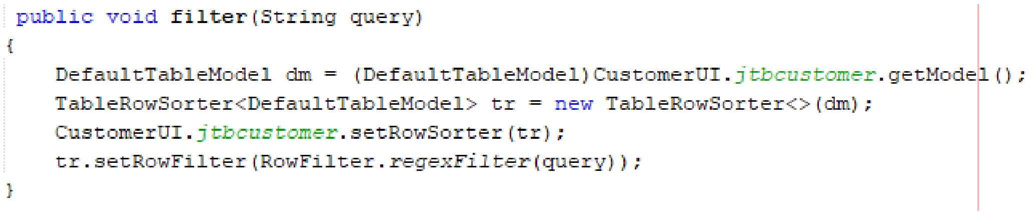
For getting the values from private variables, Accessors methods are used.



```
public String getPhonenum() {
    return Phonenum;
}

public String getAddress() {
    return Address;
}
```

Figure 7 accessors methods

Also, there are some private methods, like in Customer.java class.



```
 public void filter(String query)
{
    DefaultTableModel dm = (DefaultTableModel)CustomerUI.jtbcustomer.getModel();
    TableRowSorter<DefaultTableModel> tr = new TableRowSorter<>(dm);
    CustomerUI.jtbcustomer.setRowSorter(tr);
    tr.setRowFilter(RowFilter.regexFilter(query));
}
```

Figure 8 private methods in class

Finally, protecting the object's variables from unauthorized adjustments with the aid of encapsulation which will cause crash to the program.

3. Inheritance:

One of the main principles of object-oriented programming (OOP) languages is inheritance. For a hierarchy of classes that share a collection of attributes and methods, this is a process where deriving a class from another class. It can be used to define various kinds of exceptions, add existing frameworks with custom logic, and even map your domain model to a database. (OOP Concept for Beginners: What is Inheritance? 2020)

As there are many types of inheritance, Hierarchical inheritance is chosen as it is a type of inheritance where a single parent or base class is inherited from more than one class. The base class is also common particularly those characteristics that are common in the parent class. According to the syntax, the child class derives or inherits all the common features in the parent class, and vice versa, the methods in the child class also occur. It can therefore be assumed that the parent class properties can be inherited by the n-number of child class or base class, and vice versa can also occur. It is also not important that it is possible to inherit only common features. It is also possible to inherit some other function.

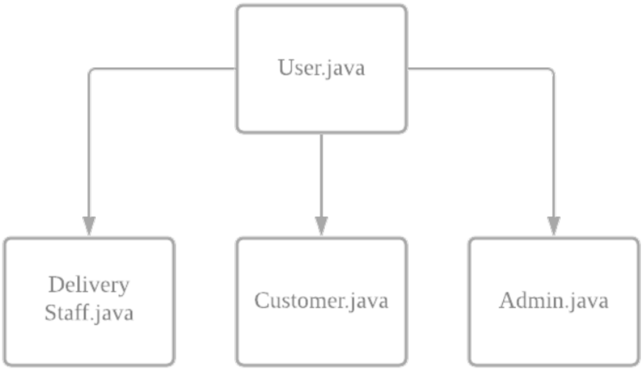
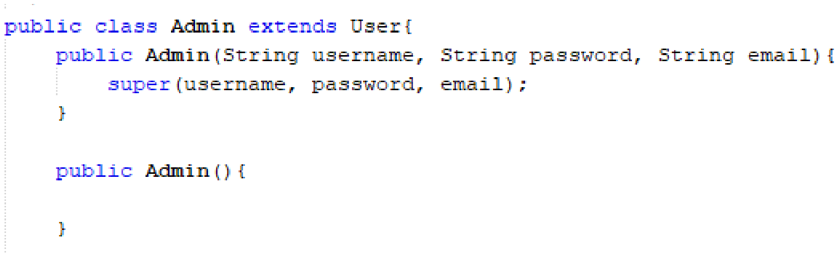


Figure 9 parent and child class

To define the class that subclass extends, the word of “Extend” is used as the extension keyword. Therefore, the class indirectly expands the Object class if it is not declared as a superclass. Object is the origin of all hierarchies of inheritance; it is the only class in Java that does not extend to another class. As an example, in the classes of Admin it has extended user as it have similar methods from User.



```
public class Admin extends User{
    public Admin(String username, String password, String email){
        super(username, password, email);
    }

    public Admin(){

    }
```

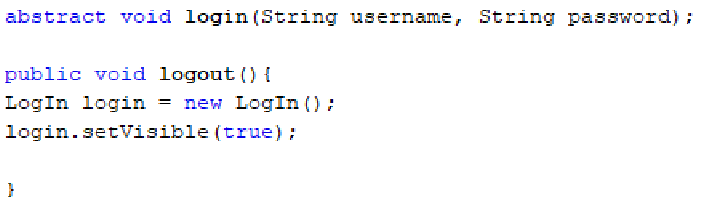
Figure 10 extend from admin

4. Polymorphism

In various ways, the term polymorphism is used and defines circumstances in which something exists in many distinct types. In computer science, the idea that objects of various types can be accessed via the same interface is defined. Each type may provide this interface with its own, independent implementation. It is one of the main object-oriented programming (OOP) principles.

Where java supports two types of polymorphism which are static and dynamic polymorphism. Static type of polymorphism allows you to implement several methods within the same class, like many other object-oriented programming languages, using the same name but a different set of parameters. With a criterion of different numbers of parameters like 1 method accepts 2 parameters or the type of parameters must be different as in 1 method accepts a String and a Long. It is expected to have different order in the parameters. (Polymorphism in Java with example, 2020) Each approach has a different signature because of the various sets of parameters. This helps the compiler to decide which method must be named and to connect it to the call of the method. This technique is referred to as static polymorphism.

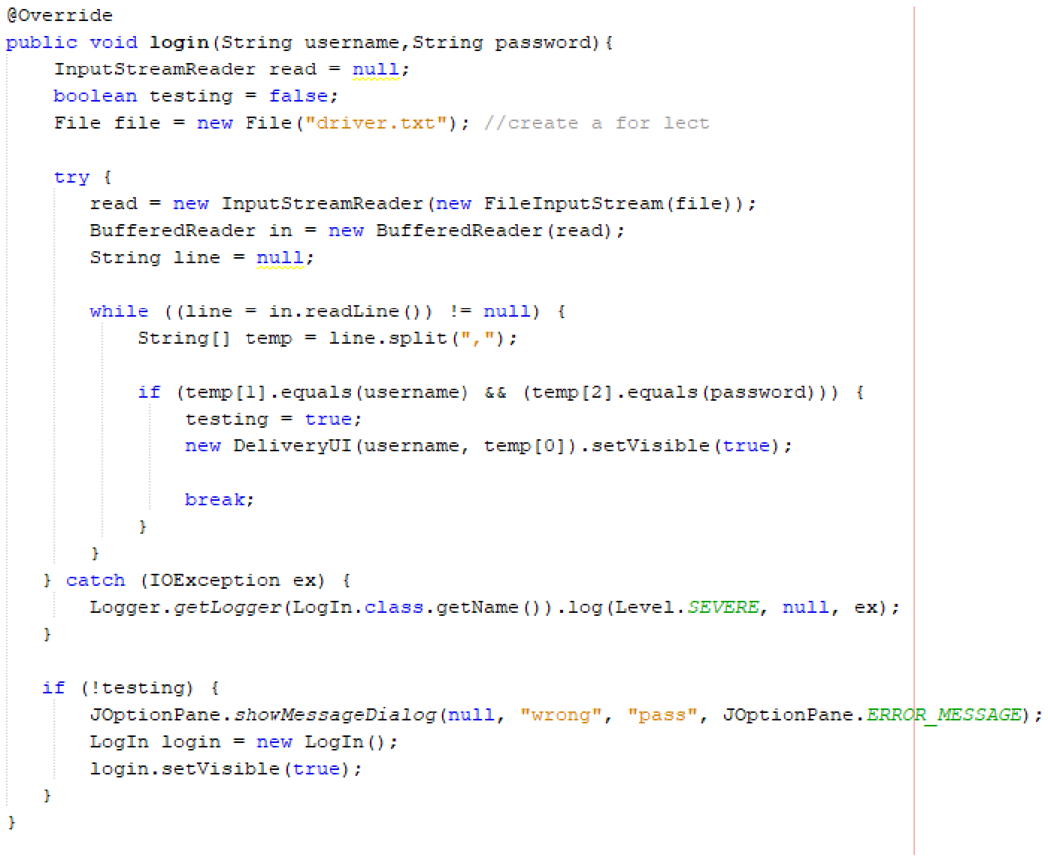
Follow by this type of polymorphism does not allow the executed method to be decided by the compiler. At runtime, the JVM needs to do that. A subclass can override a technique of its superclass within an inheritance hierarchy. That allows the subclass developer to customize or fully replace the method's actions. It produces a type of polymorphism as well. The same name and parameters are shared by both methods introduced by the super- and subclass but have different features.



```
abstract void login(String username, String password);

public void logout(){
LogIn login = new LogIn();
login.setVisible(true);

}
```

Figure 11 login in user class

As a conclusion, one of the main principles in OOP languages is polymorphism. The idea that different classes can be used with the same interface is explained. Each of these groups will supply the interface with its own implementation. Hence based on our project, dynamic polymorphism is used as there are super class and sub class like User.java and DeliveryStaff.java which shows the overridden method.



```
@Override
public void login(String username,String password){
    InputStreamReader read = null;
    boolean testing = false;
    File file = new File("driver.txt"); //create a for lect

    try {
       read = new InputStreamReader(new FileInputStream(file));
       BufferedReader in = new BufferedReader(read);
       String line = null;

       while ((line = in.readLine()) != null) {
           String[] temp = line.split(",");

           if (temp[1].equals(username) && (temp[2].equals(password))) {
               testing = true;
               new DeliveryUI(username, temp[0]).setVisible(true);

               break;
           }
       }
   } catch (IOException ex) {
       Logger.getLogger(LogIn.class.getName()).log(Level.SEVERE, null, ex);
   }

   if (!testing) {
       JOptionPane.showMessageDialog(null, "wrong", "pass", JOptionPane.ERROR_MESSAGE);
       LogIn login = new LogIn();
       login.setVisible(true);
   }
}
```

Figure 12 login in delivery class

Advantage of using OOPS

- Re-Usable

This means reusing such equipment rather than installing it over and over again. Through the use of a class, this is achieved. As per our need, we can use this 'n' number of times. As an example, the code can be reused in other class.

- Redundancy of Data

This is a state generated at the location of data storage (databases) where two different locations hold the same piece of data. So, one of the main benefits of OOP is data redundancy. If a user wants a similar feature in multiple classes, he or she can go ahead by writing and inheriting common class definitions for the similar features.

- Improved efficiency

The above-mentioned facts of the application undoubtedly increase the overall efficiency of its users. This leads to more work being done a better application being completed, more functionality implemented and easier to read, compose and maintain. To create entirely new program, an OOP programmer can stitch new software objects. This is made possible by a good number of libraries with valuable functions in abundance.

- Flexibility of Polymorphism

If the place or situation gets changed, you function in a different way. If a person is in a business, the same person would act like a student if he is in a school and if placed in a house as a son/daughter. A person will behave like a customer. Here we can see that every time the atmosphere is modified, the same individual displays distinct behavior. This implies that versatility is polymorphism, which benefits developers in a variety of ways.

Disadvantage of using OOP

- Steep curve of learning:

The phase of thinking involved in object-oriented programming. For certain individuals, it may not be normal, and it may take time to get used to it. It's difficult to build program focused on object interaction. Some of the main ones. It is possible to program methods, such as inheritance and polymorphism can be challenging to learn.

- Larger program size and slow down programs:

As OOP concept would definitely need more coding applies than procedural programs which makes the size or program bigger and would slow down the program when more instructions is instructed to be executed.

User Manual

Basically, the application has three different access which are Customer, Admin, Delivery Staff. Each one of them has its own menu and functions. The applications flow begins with the login page then follow by admin interface, admin functions, delivery staff interface, delivery staff function and customer interface.

Login

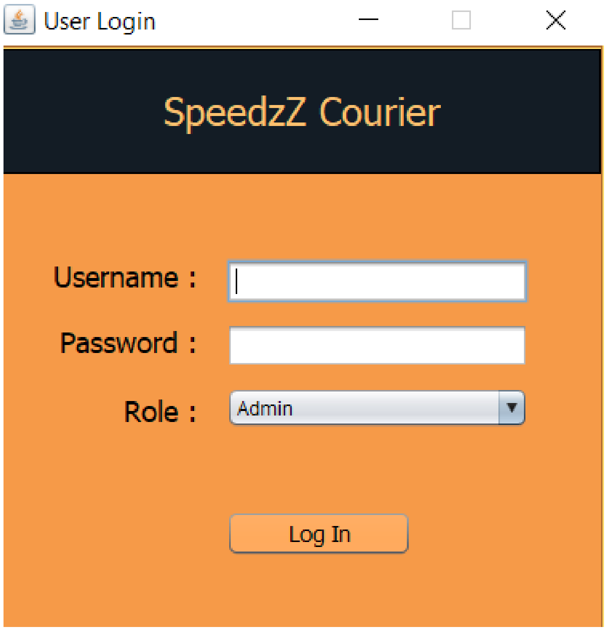
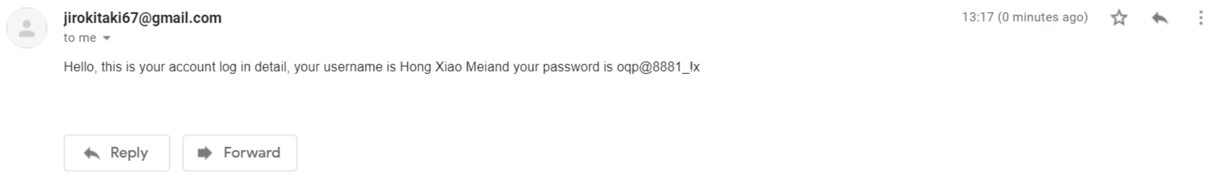


Figure 13 user login

As for the login page, user will have to log on with their username and password then choose their role. For customer to make registration to their account, they can make register based on their first order then customer will receive an email once staff have succeeded created their account.


jirokitaki67@gmail.com

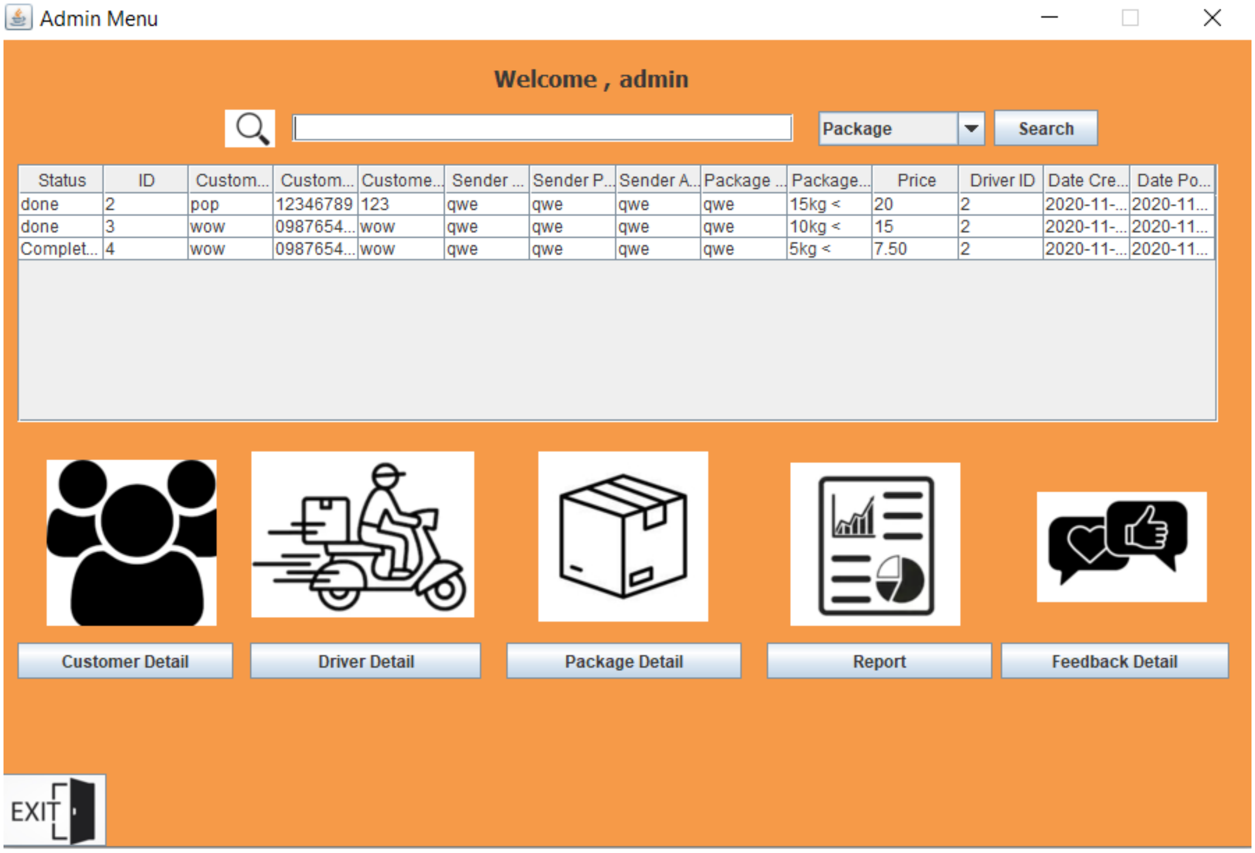
to me

13:17 (0 minutes ago)

Hello, this is your account log in detail, your username is Hong Xiao Meiand your password is oqp@8881_!x

Figure 14 registration email

Admin Menu

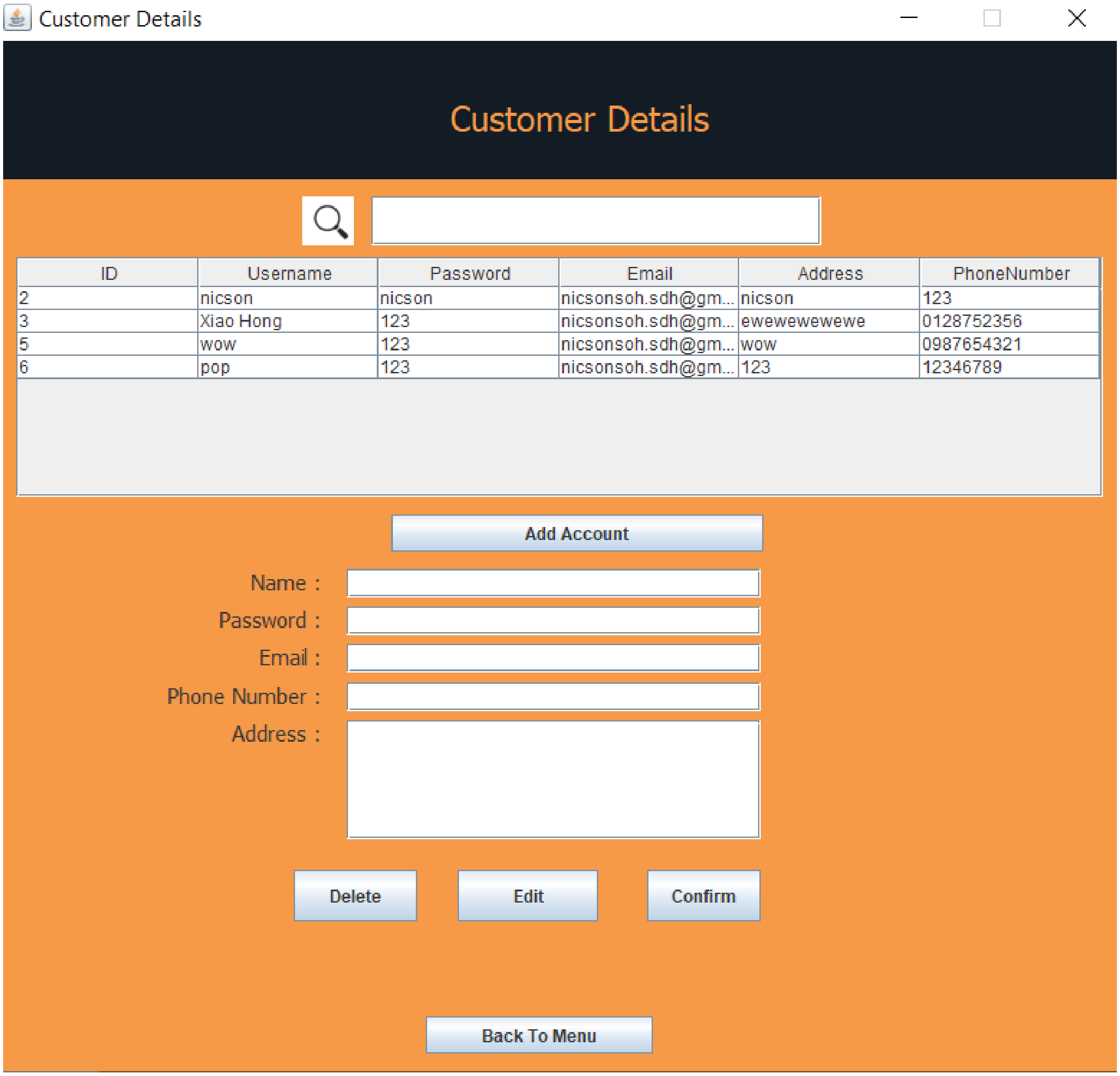


Status	ID	Custom...	Custom...	Custome...	Sender ...	Sender P...	Sender A...	Package ...	Package...	Price	Driver ID	Date Cre...	Date Po...
done	2	pop	12346789	123	qwe	qwe	qwe	qwe	15kg <	20	2	2020-11-...	2020-11...
done	3	wow	0987654...	wow	qwe	qwe	qwe	qwe	10kg <	15	2	2020-11-...	2020-11...
Complet...	4	wow	0987654...	wow	qwe	qwe	qwe	qwe	5kg <	7.50	2	2020-11-...	2020-11...

Figure 15 admin menu

After login, admin menu will show, and the login interface will be closed. Where the "Welcome Admin" will track based on the admin username that is logged into. For admin, there are a few functions which were customer details, driver details and package details

Customer Details



ID	Username	Password	Email	Address	PhoneNumber
2	nicson	nicson	nicsonsoh.sdh@gm...	nicson	123
3	Xiao Hong	123	nicsonsoh.sdh@gm...	eweweweweewe	0128752356
5	wow	123	nicsonsoh.sdh@gm...	wow	0987654321
6	pop	123	nicsonsoh.sdh@gm...	123	12346789

Figure 16 customer details

As for the customer detail page, admin can make changes onto the customer details such as email, phone number and address. Where name and password are non-editable. When admin would like to make changes onto customer information, admin will have to pressed on edit button to make edit function and confirm button to update the table. While delete button is used to delete the user account.

Add Account

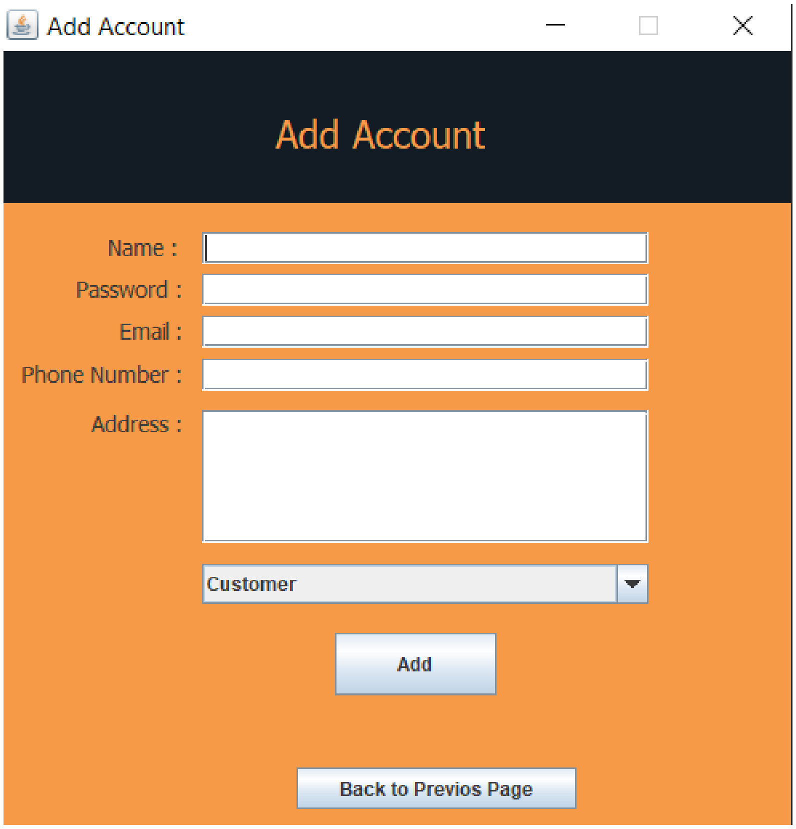
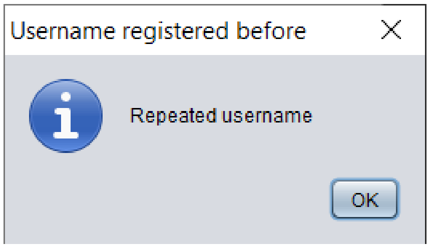


Figure 17 add account

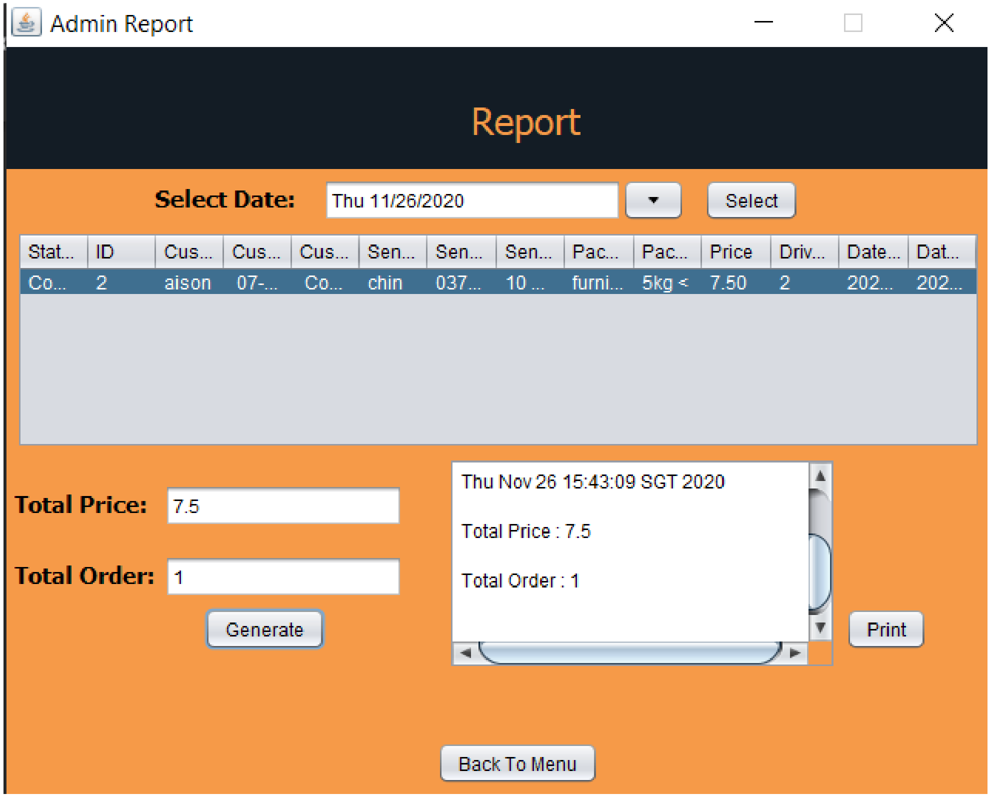

Username registered before

Repeated username

Figure 18 check system data

While admin can add customer account, admin will help new user to fill in their information and once the Add button is pressed, an email will be generated and send to the email that the account is using. When the username is repeated an error message will be shown to inform that the username has been used and new username is required to create a new account.

Admin Report



Select Date: Thu 11/26/2020

Stat...	ID	Cus...	Cus...	Cus...	Sen...	Sen...	Sen...	Pac...	Pac...	Price	Driv...	Date...	Dat...
Co...	2	aison	07-...	Co...	chin	037...	10 ...	furni...	5kg <	7.50	2	202...	202...

Total Price: 7.5

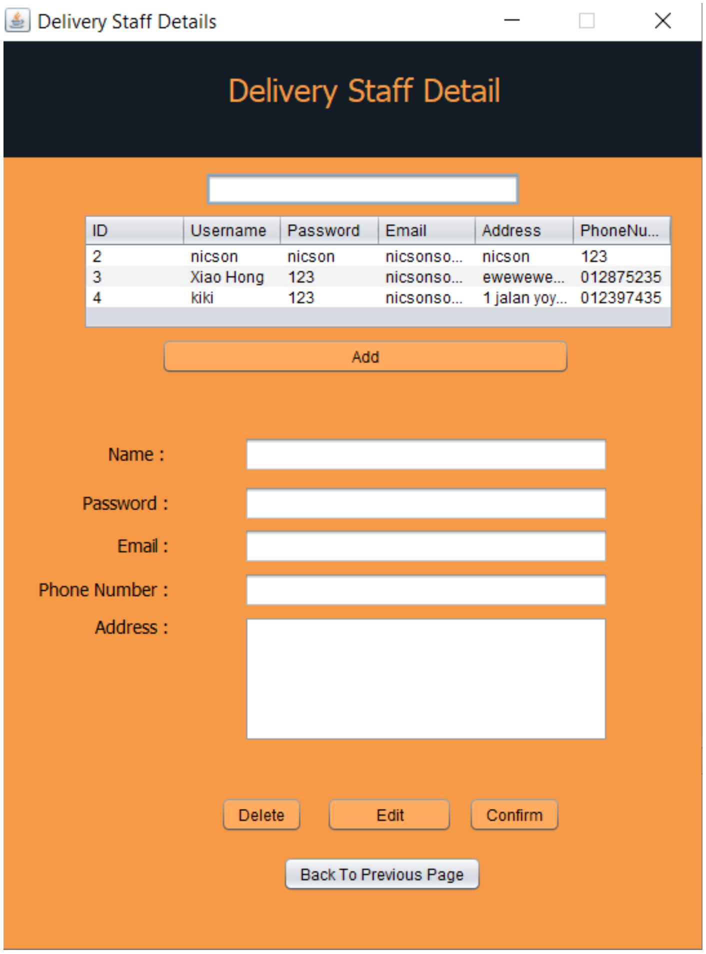
Total Order: 1

```
Thu Nov 26 15:43:09 SGT 2020

Total Price : 7.5

Total Order : 1
```

Figure 19 admin report

Admin can also generate a report by searching the date and choosing the order then generate the report to print for document purposes.

Delivery Staff Details

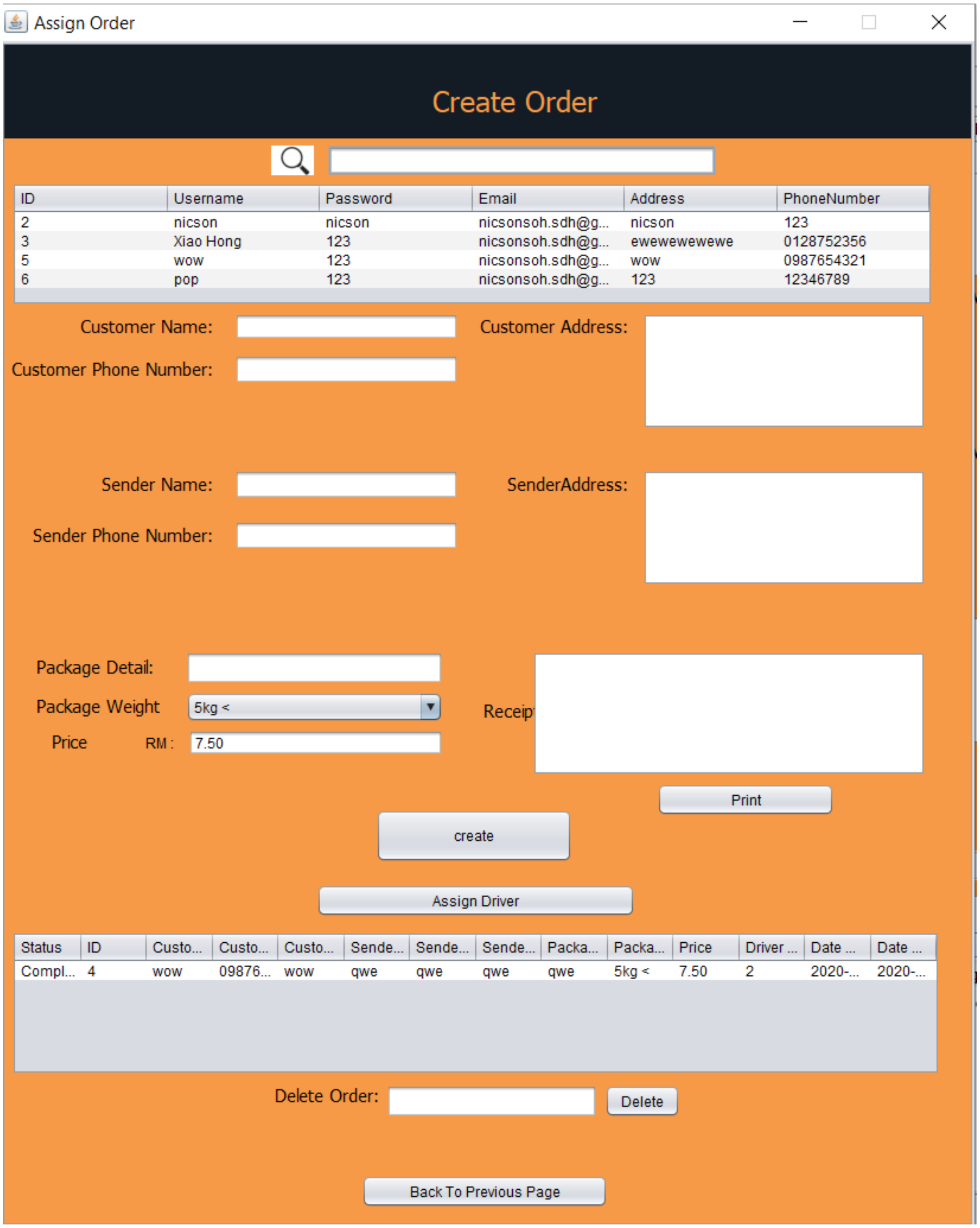


ID	Username	Password	Email	Address	PhoneNu...
2	nicson	nicson	nicsonso...	nicson	123
3	Xiao Hong	123	nicsonso...	ewewewe...	012875235
4	kiki	123	nicsonso...	1 jalan yoy...	012397435

Figure 20 delivery staff details

Delivery staff details is the GUI that shows the Delivery staff details while admin can make edit to staff accounts too by making edit, updating details and delete staff.

Create Order



ID	Username	Password	Email	Address	PhoneNumber
2	nicson	nicson	nicsonsoh.sdh@g...	nicson	123
3	Xiao Hong	123	nicsonsoh.sdh@g...	ewewewewewe	0128752356
5	wow	123	nicsonsoh.sdh@g...	wow	0987654321
6	pop	123	nicsonsoh.sdh@g...	123	12346789

Customer Name:

Customer Address:

Customer Phone Number:

Sender Name:

SenderAddress:

Sender Phone Number:

Package Detail:

Package Weight: 5kg <

Price RM : 7.50

Receip

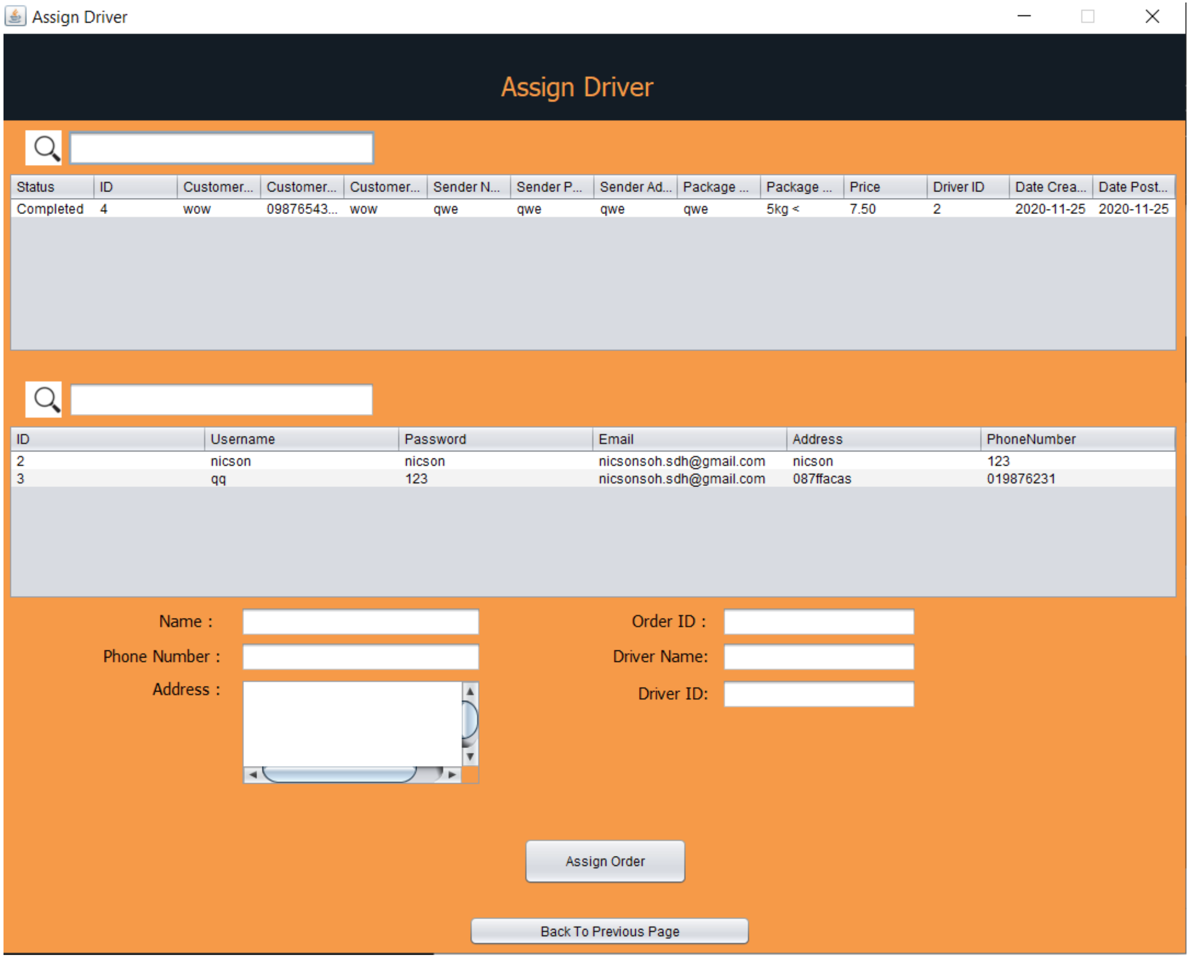
Status	ID	Custo...	Custo...	Custo...	Sende...	Sende...	Sende...	Packa...	Packa...	Price	Driver ...	Date ...	Date ...
Compl...	4	wow	09876...	wow	qwe	qwe	qwe	qwe	5kg <	7.50	2	2020-...	2020-...

Delete Order:

Figure 21 create order

As for create order, admin can add order by taking data from table or search for user from table. Then, admin will have to fill in the sender details. After filling the sender's data, the package details must be filled to proceed to the following steps. Then, a print function is including for admin to print the receipt for user. After that, the order will be shown on the second table which is the order table.

Assign Order



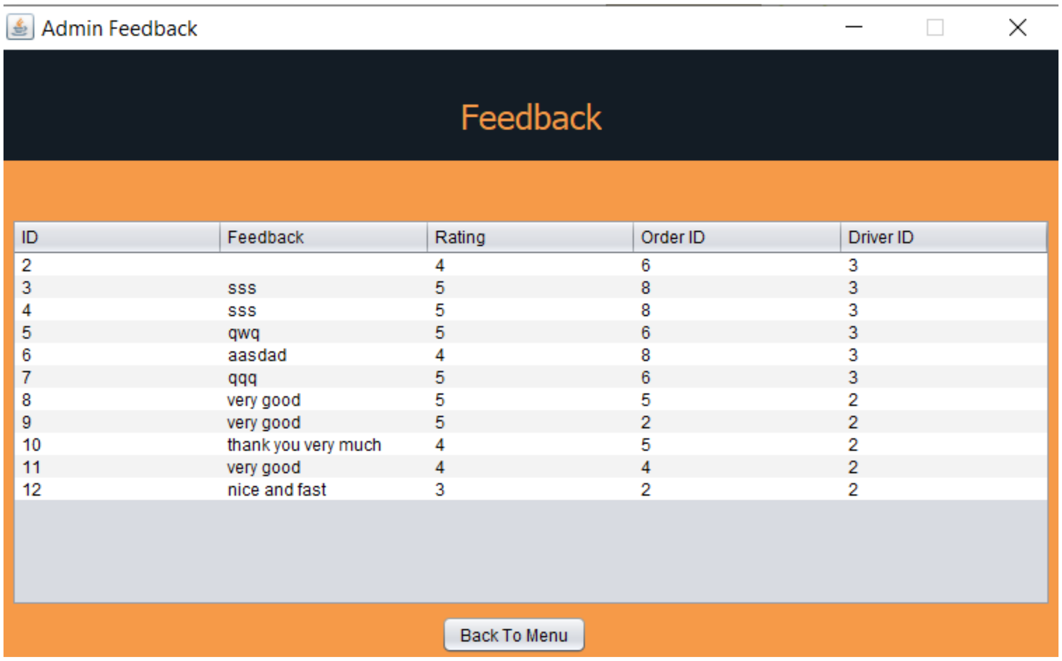
Status	ID	Customer...	Customer...	Customer...	Sender N...	Sender P...	Sender Ad...	Package ...	Package ...	Price	Driver ID	Date Crea...	Date Post...
Completed	4	wow	09876543...	wow	qwe	qwe	qwe	qwe	5kg <	7.50	2	2020-11-25	2020-11-25

ID	Username	Password	Email	Address	PhoneNumber
2	nicson	nicson	nicsonsoh.sdh@gmail.com	nicson	123
3	qq	123	nicsonsoh.sdh@gmail.com	087ffacas	019876231

Figure 22 assign order

After create order, admin will have to assign the order to the delivery staff. Once the order has been assigned, the data will be transferred to delivery staff interface to deliver staff and the data will be removed from the current event.

Admin Feedback

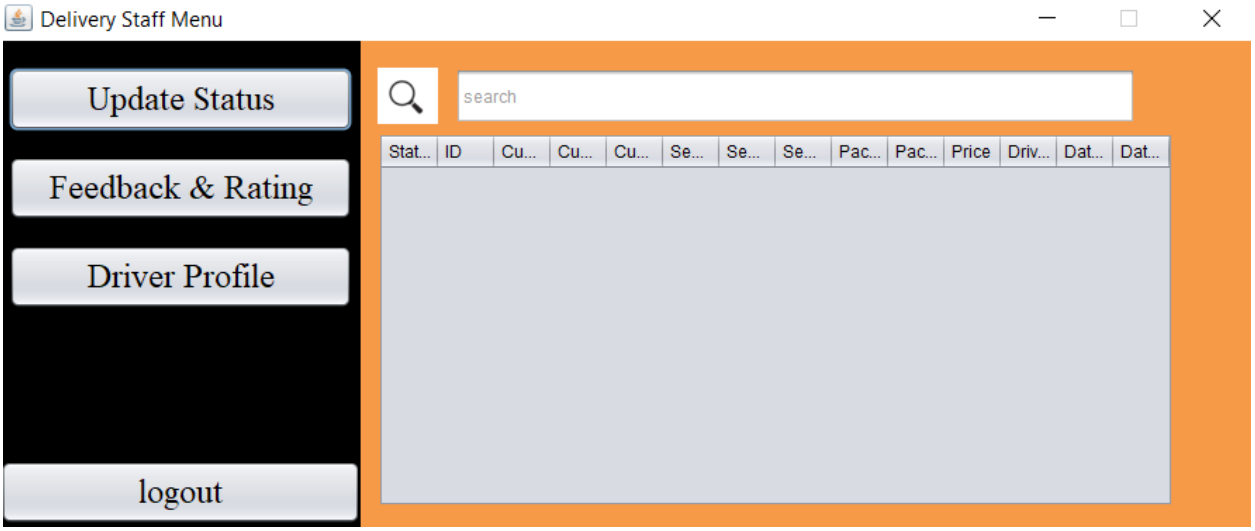


ID	Feedback	Rating	Order ID	Driver ID
2		4	6	3
3	sss	5	8	3
4	sss	5	8	3
5	qwq	5	6	3
6	aasdad	4	8	3
7	qqq	5	6	3
8	very good	5	5	2
9	very good	5	2	2
10	thank you very much	4	5	2
11	very good	4	4	2
12	nice and fast	3	2	2

Figure 23 admin feedback

Admin can view all the feedback that have been made will be shown at the interface of the admin site to have an overview of feedback of the delivery staff.

Delivery Staff Menu



Stat...	ID	Cu...	Cu...	Cu...	Se...	Se...	Se...	Pac...	Pac...	Price	Driv...	Dat...	Dat...

Figure 24 delivery staff menu

As for the delivery staff page, they have their Delivery staff menu which shows the order that currently on track. Delivery staff can search the information of order through the search bar to help out customer.

Delivery Update

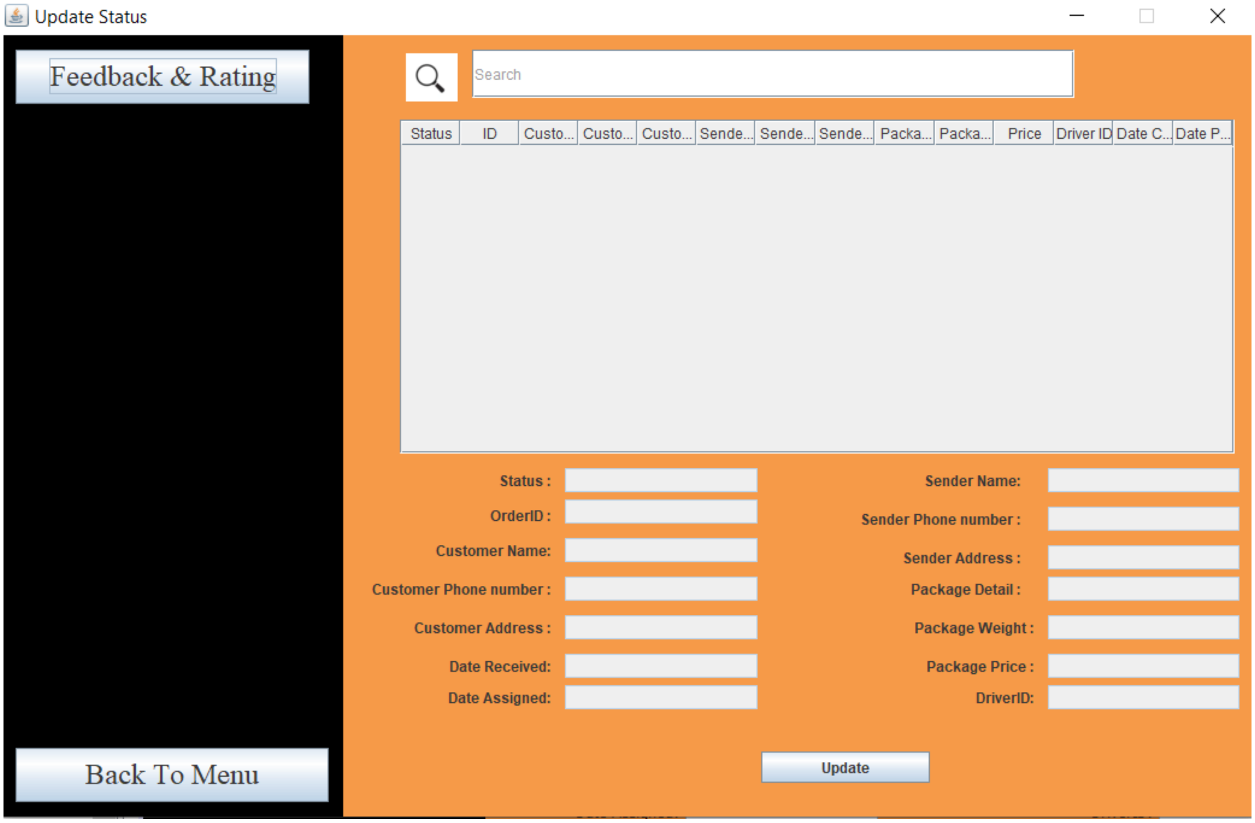
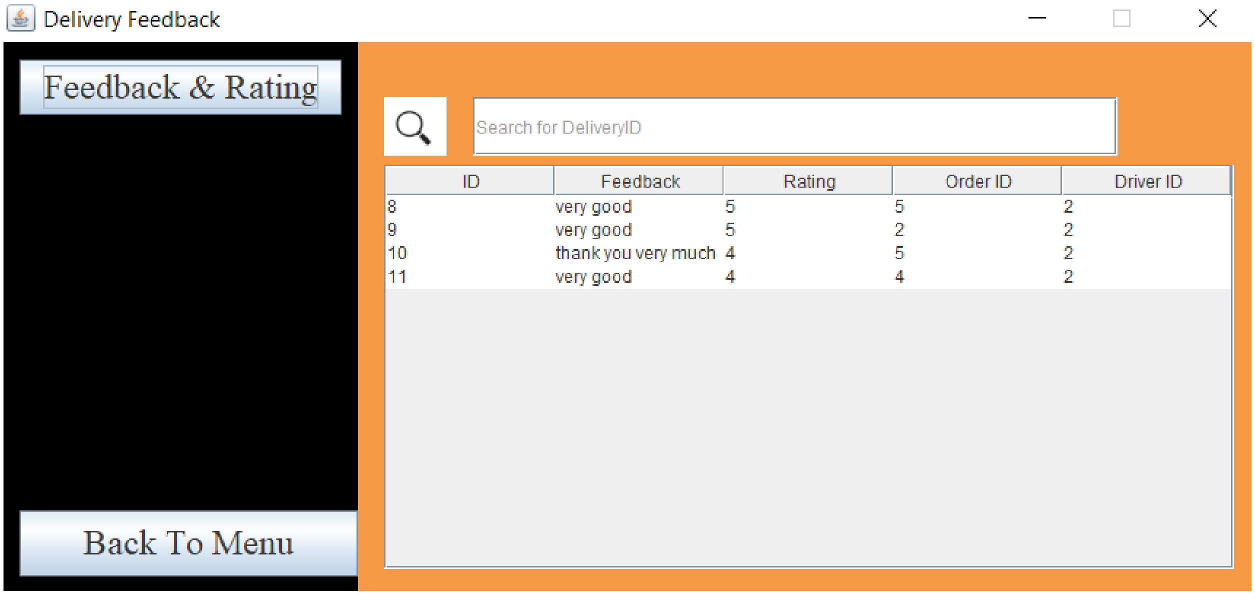


Figure 25 delivery update

As for the delivery update status, the delivery staff will have to update the date received when the order have been successfully sent. Once the order has been successfully updated, an email will send to the customer to inform the customer about the order have been successfully sent and received. When the update button is pressed, the date received will automatically update the current date to the database and change the status from pending to done.

Delivery Feedback

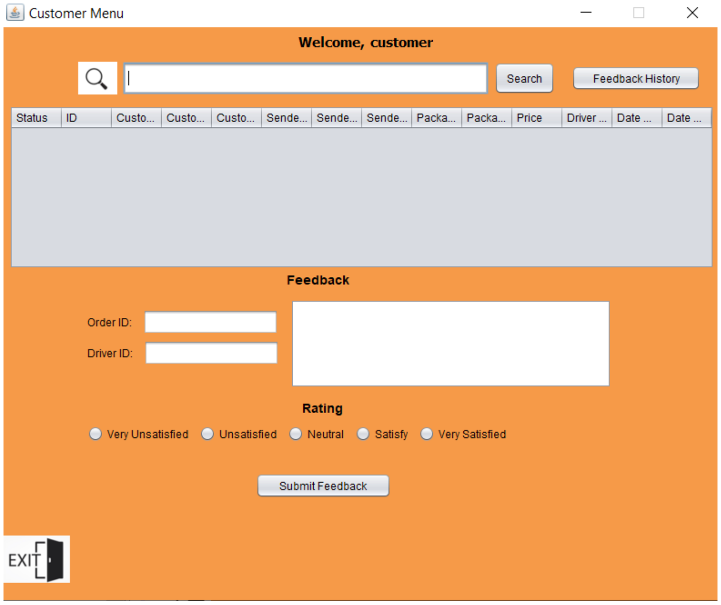


ID	Feedback	Rating	Order ID	Driver ID
8	very good	5	5	2
9	very good	5	2	2
10	thank you very much	4	5	2
11	very good	4	4	2

Figure 26 delivery feedback

Lastly, the delivery feedback which is filled by the customer is shown in this page. Where delivery staff can review the feedback from customer about the order that have been delivered.

Customer Menu

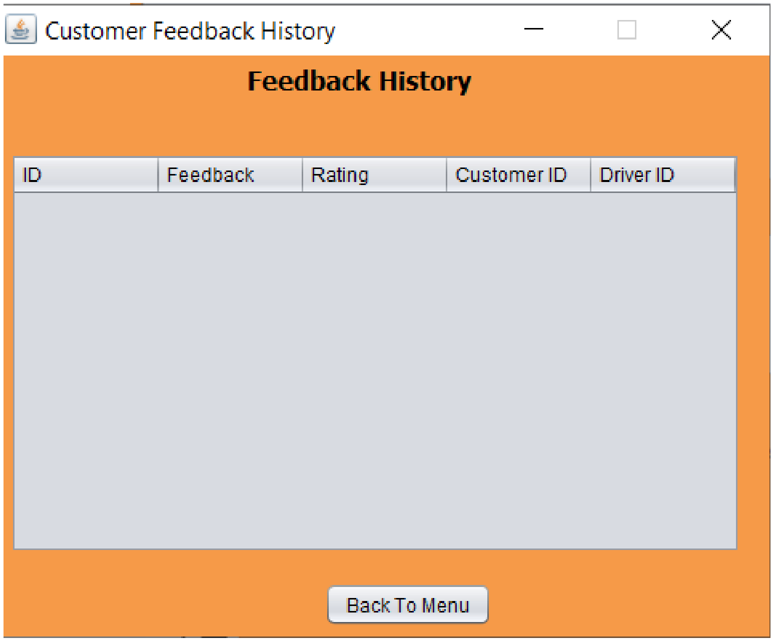


Status	ID	Custo...	Custo...	Custo...	Sende...	Sende...	Sende...	Packa...	Packa...	Price	Driver ...	Date ...	Date ...

Figure 27 customer menu

Customer menu which shows the customer interface where user can make feedback after the order have been completed.

Customer Feedback History



ID	Feedback	Rating	Customer ID	Driver ID

Figure 28 customer feedback history

As customer can view on their previous feedback on their own interface. Based on their own customer ID where the order that is already completed.

Conclusion

As a conclusion, the system was prepared with some basic requirements due to the inability to add more features into the courier as due to the shortage of time, I have managed to complete the basic requirements. A look into the future, the development team will have to improve the courier system by making it online based and have a better database system like MyphpAdmin to store the database online so that this system won’t be restricted to only for offline.

References

- OOP, A., 2020. *Advantages Of OOP | Explore The Top 9 Advantages OF OOP*. [online] EDUCBA. Available at: <https://www.educba.com/advantages-of-oop/> [Accessed 26 November 2020].
- Stackify. 2020. *What Are OOP Concepts In Java? 4 Primary Concepts*. [online] Available at: <https://stackify.com/oops-concepts-in-java/#:~:text=OOP%20concepts%20in%20Java%20are,encapsulation%2C%20inheritance%2C%20and%20polymorphism.&text=Basically%2C%20Java%20OOP%20concepts%20let,of%20them%20without%20compromising%20security.> [Accessed 26 November 2020].
- GeeksforGeeks. 2020. *Classes And Objects In Java - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/classes-objects-java/> [Accessed 26 November 2020].
- Braunschweig, D., 2020. *Encapsulation*. [online] Press.rebus.community. Available at: <https://press.rebus.community/programmingfundamentals/chapter/encapsulation/#:~:text=Encapsulation%20is%20one%20of%20the,parties'%20direct%20access%20to%20them.> [Accessed 26 November 2020].
- Stackify. 2020. *OOP Concept For Beginners: What Is Inheritance?* [online] Available at: <https://stackify.com/oop-concept-inheritance/> [Accessed 26 November 2020].
- beginnersbook.com. 2020. *Polymorphism In Java With Example*. [online] Available at: <https://beginnersbook.com/2013/03/polymorphism-in-java/> [Accessed 26 November 2020].

1)50%

Group Member Name/ID: 1)hong jun khuan

2)soh de hui

TOTAL MARKS

2)50%

1)50%

2)50%

TOTAL MARKS

Learning Outcomes	Questions
1. **Explain the techniques of Object Oriented Design(C2,PLO1)**	Class Test
2. **Implement a software application that exploits the strength of object-oriented paradigm(C6,PL02)**	**LO covered in the assignment**
3. **Demonstrate the use of object oriented concepts and their functionalities in the existing system(A3,PL04)**	**LO covered in the assignment**

Assignment Question	**Cognitive Level**						**Psychomotor Level**							**Affective Level**				
	1	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**1**	**2**	**3**	**4**	**5**
						90M										**10M**		
OOMS						**90%**										**10%**		

REQUIREMENT ANALYSIS [CLO2-PLO2]	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0-3	**4**	**5-6**	**7**	**8-10**
USE CASE DIAGRAM WITH DESCRIPTION (10%) [CLO2]	Incorrect overview use case, detail use cases with scenarios (where applicable) and use case descriptions OR incorrect user case notations OR illogical functional design.	Inappropriate overview use case, detail use cases with scenarios (where applicable) and use case descriptions.	The use cases are well presented (generalization) with no major mistake in logic and notation, and described all flows in use case descriptions.	Appropriate labelling and no mistake in logic and notation and clear description for normal flow, subflow and alternative flow in use case descriptions.	Comprehensive provision of the required - overview use case; - detail use case diagrams with scenarios (where applicable: generalization, extends, includes); - use case descriptions for each use case / scenario and no mistake in logic and notation and appropriateness.

CLASS DIAGRAM (10%) [CLO2 and CLO3]	No attributes and no associations are included.	Class diagram with attributes and associations. Both are incomplete and illogical.	Class diagram with attributes and associations. Both are complete with no major errors.	Class diagram with appropriate attributes and associations. With labelling and no mistakes in logic and notation.	Class diagram with appropriate, relevant attributes and associations. With very good and meaningful labelling according to guidelines.

IMPLEMENTATION [CLO2-PLO2]	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0-7	**8-9**	**10-12**	**13-14**	**15-20**
***Group Member A* USER MANAGEMENT (20%)**	No program or work not done. Program incomplete with obvious errors. Not able to compile and run the program.	Not able to compile or run but evidence of the coding is available. Able to compile but not able to run the program. Able to compile and run the program but only able to add less than 50% of the details of access rights module listed in the assignment. Not able to demonstrate the use of object-oriented concepts. Data not stored in file.	Able to compile and run the program. Able to add at least 50% of the details of user access module listed in the assignment. Able to demonstrate the use of at least three of the object-oriented concepts – such as creation of classes, objects and methods. Attempted to write to file but with some errors.	Able to compile and run the program. Able to add at least 80% of the details of access rights module listed in the assignment. Able to demonstrate most of the object-oriented concepts. Able to write to file and read from file with some errors.	Able to compile and run the program. Able to add 90-100% of the details of access rights module listed in the assignment. Able to demonstrate all the object-oriented concepts along with additional features. Able to write to file and read from file with no errors.
***Group Member A* REPORT MANAGEMENT (20%)**	No program or work not done. Program incomplete with obvious errors. Not able to compile and run the program.	Not able to compile or run but evidence of the coding is available. Able to compile but not able to run the program. Able to compile and run the program but only able to add less than 50% of the details of module listed in the assignment. Not able to demonstrate the use of object-oriented concepts. Data not stored in file.	Able to compile and run the program. Able to add at least 50% of the details of module listed in the assignment. Able to demonstrate the use of at least three of the object-oriented concepts – such as creation of classes, objects and methods. Attempted to write to file but with some errors.	Able to compile and run the program. Able to add at least 80% of the details of the module listed in the assignment. Able to demonstrate most of the object-oriented concepts. Able to write to file and read from file with some errors.	Able to compile and run the program. Able to add 90-100% of the details of the module listed in the assignment. Able to demonstrate all the object-oriented concepts along with additional features. Able to write to file and read from file with no errors.
***Group Member B* ORDER MANAGEMENT (20%)**	No program or work not done. Program incomplete with obvious errors. Not able to compile and run the program.	Not able to compile or run but evidence of the coding is available. Able to compile but not able to run the program. Able to compile and run the program but	Able to compile and run the program. Able to add at least 50% of the details of module listed in the assignment. Able to	Able to compile and run the program. Able to add at least 80% of the details of the module listed in the assignment. Able to	Able to compile and run the program. Able to add 90-100% of the details of the module listed in the assignment. Able to

		only able to add less than 50% of the details of module listed in the assignment. Not able to demonstrate the use of object-oriented concepts. Data not stored in file.	demonstrate the use of at least three of the object-oriented concepts – such as creation of classes, objects and methods. Attempted to write to file but with some errors.	demonstrate most of the object-oriented concepts. Able to write to file and read from file with some errors.	demonstrate all the object-oriented concepts along with additional features. Able to write to file and read from file with no errors.
Group Member B **DELIVERY MANAGEMENT (20%)**	No program or work not done. Program incomplete with obvious errors. Not able to compile and run the program.	Not able to compile or run but evidence of the coding is available. Able to compile but not able to run the program. Able to compile and run the program but only able to add less than 50% of the details of module listed in the assignment. Not able to demonstrate the use of object-oriented concepts. Data not stored in file.	Able to compile and run the program. Able to add at least 50% of the details of module listed in the assignment. Able to demonstrate the use of at least three of the object-oriented concepts – such as creation of classes, objects and methods. Attempted to write to file but with some errors.	Able to compile and run the program. Able to add at least 80% of the details of the module listed in the assignment. Able to demonstrate most of the object-oriented concepts. Able to write to file and read from file with some errors.	Able to compile and run the program. Able to add 90-100% of the details of the module listed in the assignment. Able to demonstrate all the object-oriented concepts along with additional features. Able to write to file and read from file with no errors.

REPORT [CLO3-PLO4]	**Fail** **0-3**	**Marginal Fail** **4**	**Pass** **5-6**	**Credit** **7**	**Distinction** **8-10**
REPORT FORMAT AND REFERENCES (10%)	The reference list is *all inapplicable* OR *irrelevant*. The format is NOT comply proper referencing system. The citation is NOT included at all. The simple document without formatting, header and footer, page number, etc.	The reference list is identified mostly inapplicable OR irrelevant, The format is partially comply proper referencing system. The citation is partially included. The simple document with little formatting.	The reference list is complete but sort of complete citation. The document is complete.	The above +. The citation is clearly specified. The above + with all the formatting criteria.	The above + with rich source of explorations to form a complete reference. The above + precise, clear, complete, all the diagram, chart, picture, pie, symbol, glossary are completely organized.
PROGRAM DOCUMENTATION (20%)	**Fail**	**Marginal Fail**	**Pass**	**Credit**	**Distinction**
	0-7	**8-9**	**10-12**	**13-14**	**15-20**
	Documentation not done. Content of documentation does not adhere to any of the requirements stipulated in the assignment requirements.	At least 1 object-oriented programming concept applied in the solution and briefly described. No implementation code for the object-oriented programming concepts identified. Screen capture of the output of the program does not have any explanation to describe the program.	Description of at least 2 object-oriented programming concepts that are applied in the solution with some evidence of the implementation code is included. Screen capture of the output of the program with minimal explanation to describe the program.	Description of at least 3 object-oriented programming concepts that are applied in the solution are described and evidence of the implementation code being documented. Screen capture of the output of the program with some explanation to describe the program. Description and evidence of at least 1 additional feature which has been incorporated in the solution.	Description of at least 4 object-oriented programming concepts that are applied in the solution and evidence of the implementation code being documented. Screen capture of the output of the program with appropriate explanation to sufficiently describe the program. Description and evidence of at least 2 additional features which have been incorporated in the solution.
INDIVIDUAL PRESENTATION (10%)	**0-3**	**4**	**5-6**	**7**	**8-10**
	Absent OR late OR not prepared for presentation session without valid reasons. Handled questions grossly bad and unable to demonstrate understanding of OO concept.	Reading from presentation material. Presentation material is NOT well prepared. Presentation sequence NOT well organized and not smooth. Handled questions badly and unable to demonstrate understanding of OO concept.	Reading occasionally from presentation material. Presentation material is well prepared. Presentation sequence acceptably organized and smooth. Handled questions well and demonstrated fundamental level of understanding of OO concept.	Good oral presentation. Presentation material is well prepared. Presentation sequence well organized and smooth. Handled questions well and demonstrated good understanding of OO concept.	Give an impactful presentation where the presenter delivers smooth oral presentation aided beautifully by well-prepared presentation material. Presentation sequence excellently planned organized and smooth. Handled questions well and demonstrated good understanding of OO concept.