

|   |     |
|---|-----|
| Campus  | NPI |
| <b>APU Foundation &amp; Degree Programmes</b> |     |



Please Tick (✓) for Re-do Assignment

### Coursework Submission and Feedback Form

|                     |  |         |                        |                                       |
|---------------------|--|---------|------------------------|---------------------------------------|
| Name                | Sandesh Giri   |         |                        | Name of Group Members (If Applicable) |
| Student No.         |  | Intake: | NPI2F1909IT            | Suraj Pandey (NPI000051)              |
| Module Code & Title | CT038-3-2 OODJ (Object Oriented Development with Java) |         |                        | Sandesh Giri (NPI000041)              |
| Assignment Title    | COURIER SERVICE SYSTEM                                 |         |                        |                                       |
| Name of Lecturer    | Mr.Sushil Adhikari                                     |         |                        |                                       |
| Date Due            | 13/08/2021   | E-Mail: | girisandy122@gmail.com |                                       |

I have read and understood the regulations on Plagiarism and Academic Dishonesty and declare that the work submitted does not breach those regulations.      Signed: \_\_\_\_\_

You must hand in to the designated APU Administrator – ensure that you receive your receipt.

| Received By | Signature | Date | Time |
|-------------|-----------|------|------|
|             |           |      |      |

| Criteria | Weighting | Fail | Marginal Fail | Pass | Credit | Distinction |
|----------|-----------|------|---------------|------|--------|-------------|
|          |           |      |               |      |        |             |

**Additional Comments:** (These may be listed below or attached)

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

**Provisional Assessment Result:** \_\_\_\_\_ **Grade** \_\_\_\_\_ **Date** \_\_\_\_\_ **Lecturer's Initial**

Taking account of above factors, the overall provisional assessment of your work is:

| <b>Distinction</b>   |           | <b>Credit</b>     |           | <b>Pass</b>      |                  |            |
|----------------------|-----------|-------------------|-----------|------------------|------------------|------------|
| A+: 80-100%          | A: 75-79% | B+: 70-74%        | B: 65-69% | C+: 60-64%       | C: 55-59%        | C-: 50-54% |
| <b>Marginal Fail</b> |           | <b>Fail</b>       |           |                  |                  |            |
| <b>D: 40-49%</b>     |           | <b>F+: 30-39%</b> |           | <b>F: 20-29%</b> | <b>F-: 0-19%</b> |            |

*The comments and assessment result are subject to both internal and external moderation at the appropriate Examination Board.*

*Consequently, they may not reflect your final grade. You may not appeal against these result on ground of Academic Judgement.*



# ASSIGNMENT

INFOMAX COLLEGE OF IT AND MANAGEMENT

**(CT038-3-2 OODJ)**

***(Object Oriented Development with Java)***

**HAND OUT DATE: (11/06/2021)**

**HAND IN DATE: (13/08/2021)**

**WEIGHTAGE: (50) %**

---

## INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment at the administrative counter.
2. Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)
3. Late submissions will be awarded zero(0) unless Extenuating Circumstances are upheld.
4. Cases of plagiarism will be penalized.
5. The assignment should be bound in appropriate style (comb bound or stapled)
6. Where the assignment should be submitted in hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hard copy.
7. You must obtain 50% overall to pass this module.

## **Acknowledgement**

This project report is prepared for the partial fulfilment of the requirement for the degree of B.Sc. IT in accordance with the rules and regulations prescribed by Asia Pacific University (APU). We would like to express our heartfelt gratitude towards APU for providing us an opportunity to apply our theoretical knowledge into the practical world to enhance our knowledge. Further, we are grateful to Mr. Sushil Adhikari, subject teacher of Object Oriented with Java for his guidance and help in completing this project. We are indebted to our teacher and friends for being a constant source of encouragement, inspiration and support.

## **Abstract**

The main purpose of this assignment is to develop an application called courier management system by implementing (Object Oriented Programming) OOP concept. OOP concept includes the use of class, object, inheritance, constructor, polymorphism. Before the development of this application class diagram is drawn which Later on helps to code this system. Along with class diagram use case diagram is also drawn to identify the features that need to be include in the system. Since the assignment has asked to store the data on txt file we have created txt file to store each and every data in this system. Similarly, the tool that is used to code this system is netbeans.

## **Table of content**

|                                       |    |
|---------------------------------------|----|
| Acknowledgement -----                 | 4  |
| Abstract -----                        | 4  |
| List of Figures -----                 | 7  |
| List of Abbreviation -----            | 7  |
| Work Load Matrix -----                | 8  |
| 1. Introduction -----                 | 9  |
| 1.1 Assumption-----                   | 9  |
| 2. Requirement Analysis-----          | 10 |
| 2.1. Class Diagram-----               | 10 |
| 2.2. Use Case Diagram -----           | 11 |
| 3. OOP and System Implementation----- | 19 |
| 3.1. Class and Object -----           | 19 |
| 3.2. Encapsulations-----              | 20 |
| 3.3. Inheritance -----                | 21 |
| 3.4. Constructor-----                 | 22 |
| 3.5 Polymorphism-----                 | 22 |
| 4. User Manual -----                  | 23 |
| 4.1. Home page-----                   | 23 |
| 4.2. Features-----                    | 24 |
| 4.2.1. Login form-----                | 24 |
| 4.2.2. Register Form -----            | 25 |
| 4.3. Managing Staff Panel-----        | 25 |
| 4.3.1. User Management -----          | 26 |
| 4.3.2. Order Management-----          | 29 |
| 4.3.3. Report Management-----         | 32 |

|                                     |    |
|-------------------------------------|----|
| 4.3.4. Feedback management-----     | 33 |
| 4.4. Customer features-----         | 34 |
| 4.5. Delivery Staff Panel -----     | 35 |
| 4.5.1. Login Form -----             | 35 |
| 4.5.2. View Order-----              | 35 |
| 4.5.3. View and Update Profile----- | 36 |
| 5. Conclusion -----                 | 37 |
| 6. References-----                  | 37 |

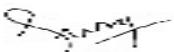
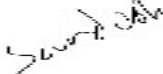
## **List of Figures**

|  |    |
|--|----|
| Figure 1: Home page-----                         | 23 |
| Figure 2: Managing Staff Login Form-----         | 24 |
| Figure 3: Managing Staff Registration Form ----- | 25 |
| Figure 4: User Account Management-----           | 26 |
| Figure 5: Customer Account Management-----       | 27 |
| Figure 6: Delivery Account Management-----       | 27 |
| Figure 7: View Delivery Person-----              | 28 |
| Figure 8: Search Delivery Account -----          | 28 |
| Figure 9: Order Management-----                  | 30 |
| Figure 10: Add Delivery-----                     | 30 |
| Figure 11: Invoice -----                         | 31 |
| Figure 12: View Delivery Status -----            | 32 |
| Figure 13: View Report-----                      | 32 |
| Figure 14: Feedback management -----             | 33 |
| Figure 15: Track Parcel-----                     | 34 |
| Figure 16: View Current Location -----           | 34 |
| Figure 17: Delivery Staff Login Form-----        | 35 |
| Figure 18: View Order-----                       | 35 |
| Figure 19: View Individual Profile-----          | 36 |
| Figure 20: View Feedback -----                   | 36 |

## **List of Abbreviation**

- OOP = Object Oriented Programming  
DBMS = Database Management System  
UI = User Interface  
N.A. = Not Available

## Work Load Matrix

|                               | <b>Sandesh Giri (NPI000041)</b>   | <b>Suraj Pandey(NPI000051)</b>  |
|-------------------------------|---|---|
| Introduction                  | 50%   | 50%   |
| Class Diagram                 | -   | 100%  |
| Use case Diagram              | 100%  | -   |
| OOP and System Implementation | 50%   | 50%   |
| User Account Management       | 50%   | 50%   |
| Order Management              | -   | 100%  |
| Feedback Management           | 100%  | -   |
| Report Management             | 100%  | -   |
| Remark                        |  |  |

## **1. Introduction**

Java is a class based object oriented programming language which was developed in 1991 by James Gosling and Patric Naughton which was later obtained by Oracle (BegginersBook,2012). This is one of the widely used programming language because it is fast, reliable, simple and can run in all platform.

In case of our assignment, we are required to develop a courier management system which allows us to manage users account, manage order, manage report. All the functionality like register, search, delete and update comes under user account management feature. The end users of this application are managing staff and delivery staff but customer can also use this application to track their parcel.

### **1.1 Assumption**

The assumption of how our program works are:

1. Managing staff can create, view, delete and update user's accounts.
2. All the information are stored in txt file as this system does not implement DBMS.
3. Delivery person can only manage his profile, view delivery status and update location.
4. With the help of parcel id customer can view their oder status.
5. Managing staff and delivery staff need to login ignorer use the power.
6. Order are add by managing staff only.

## 2. Requirement Analysis

### 2.1. Class Diagram

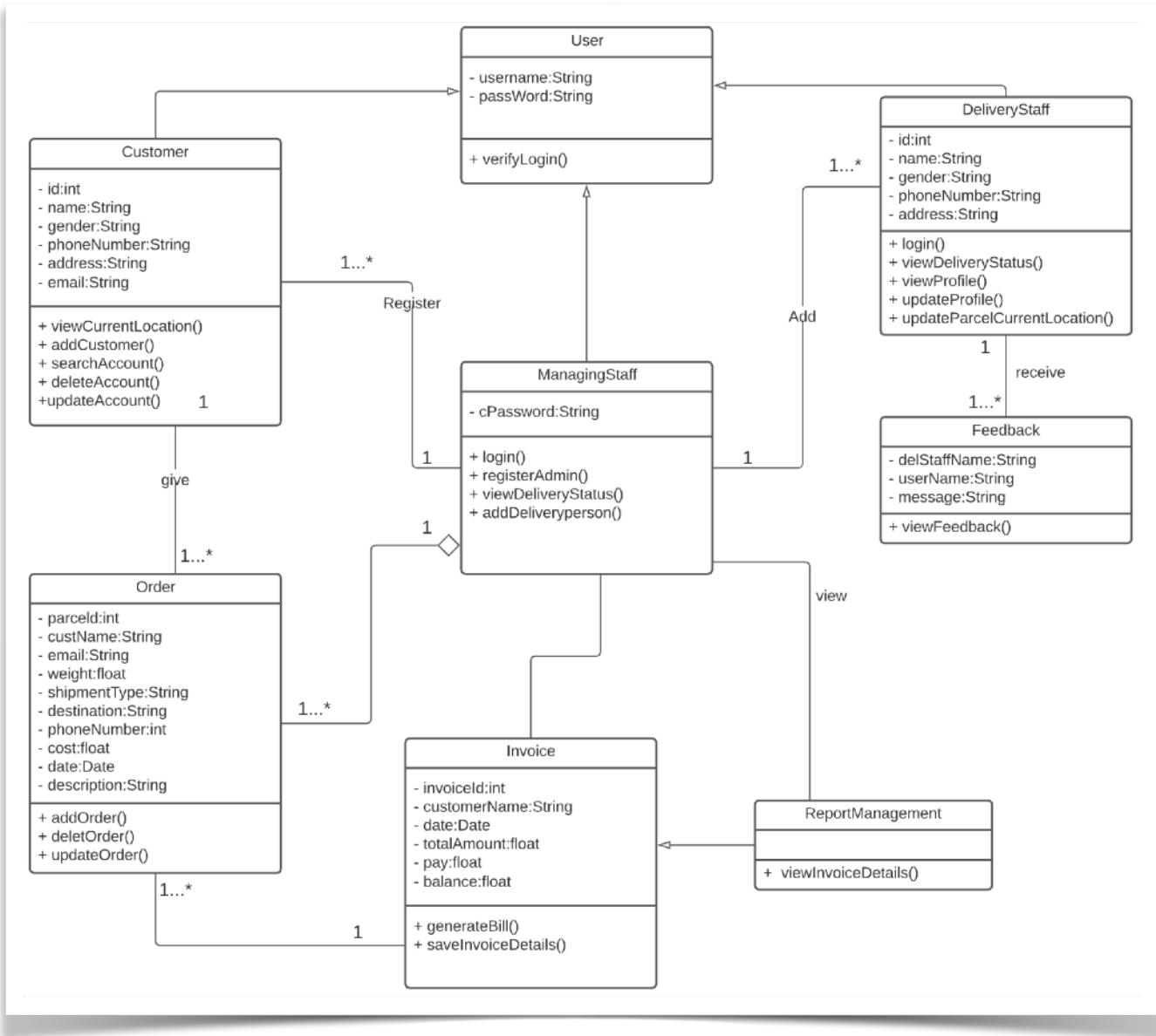


Figure : class diagram of ‘courier management System’

## 2.2. Use Case Diagram

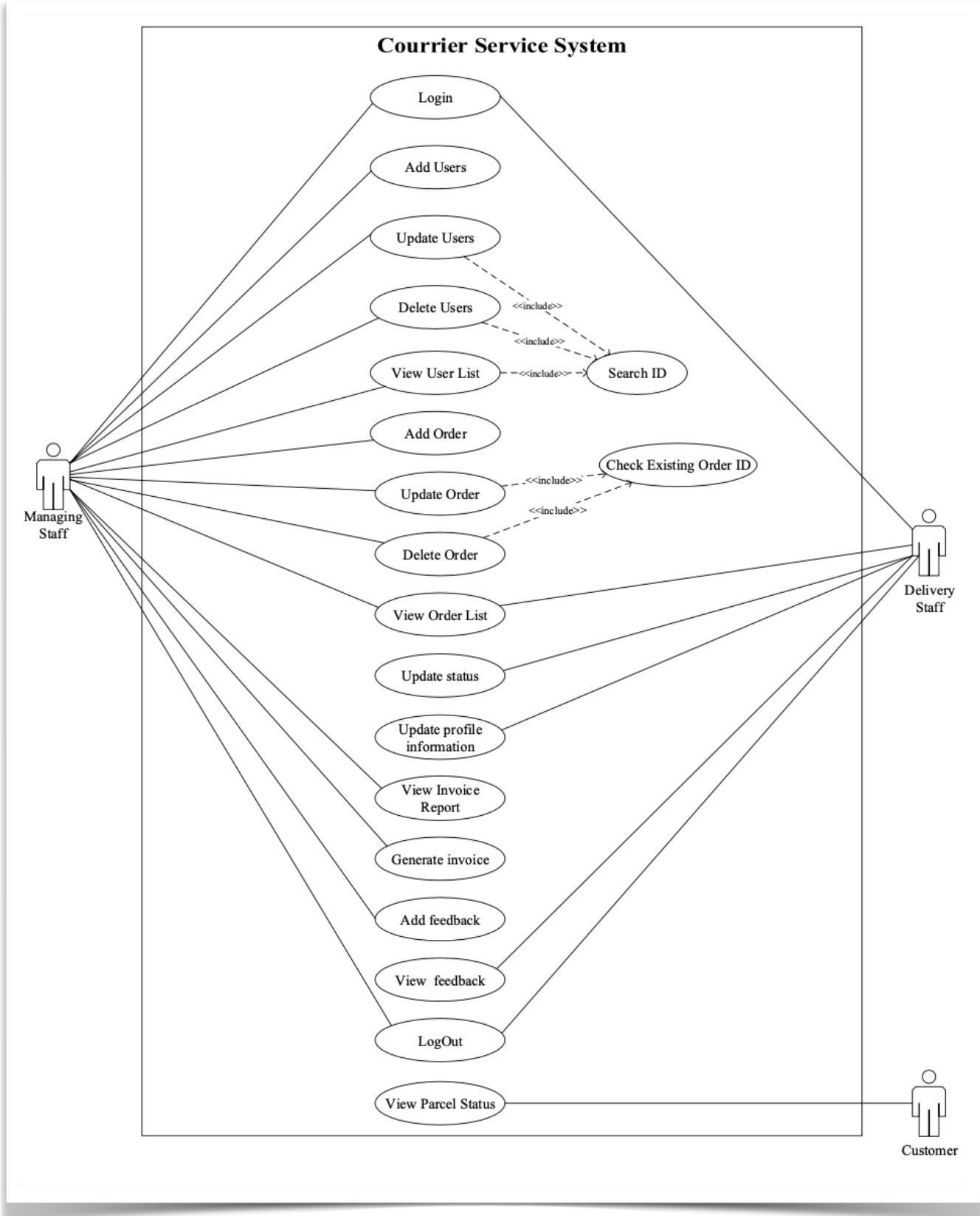


Figure :Use case diagram of ‘courier management system’

## **Use Case Specification:**

|                    |   |
|--------------------|---|
| Use Case:          | <b>Login</b>  |
| Brief Description: | Only registered users are permitted to login into the system  |
| Actors             | Managing staff and Delivery staff   |
| Precondition       | Delivery staffs are to be authorized by a managing staff before they access the system.   |
| Main Flow          | <ul style="list-style-type: none"> <li>i. User needs to enter their username and password to access the system.</li> <li>ii. The information of user will be checked and verified by the system. After that, the system will allow the users to login based on their specific roles.</li> <li>iii. End of use case</li> </ul> |
| Alternative flow   | <p>Invalid detail</p> <ul style="list-style-type: none"> <li>• System will display the error message of wrong username/password.</li> <li>• Users re-enter their login information.</li> </ul>  |

|                    |   |
|--------------------|---|
| Use Case:          | <b>Update status</b>  |
| Brief Description: | This allows the delivery staff to update the customer information and current location of the order placed by customer.   |
| Actors             | Delivery Staff  |
| Precondition       | Delivery staff needs to be logged in as a user and will update the status when the parcel is arrived.   |
| Main Flow          | <ul style="list-style-type: none"> <li>i. Delivery staff will update the status to complete if the item is sent successfully to customer.</li> <li>ii. End of use case</li> </ul> |

|                           |   |
|---------------------------|---|
| <b>Use Case:</b>          | <b>Add user</b>   |
| <b>Brief Description:</b> | This allows the managing staff to add or create account of the user.  |
| <b>Actors</b>             | Managing staff  |
| <b>Precondition</b>       | Managing staff needs to be logged in as a user  |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff will fill the essential information of the new user.</li> <li>ii. Managing staff will fill all the section.</li> <li>iii. End of use case</li> </ul> |
| <b>Alternative flow</b>   | Phone number of user is wrongly given and details could not send through message to user. So, managing staff needs to re-enter correct phone number.  |

|                           |   |
|---------------------------|---|
| <b>Use Case:</b>          | <b>Delete user</b>  |
| <b>Brief Description:</b> | This allows the managing staff to delete account of the user.   |
| <b>Actors</b>             | Managing staff  |
| <b>Precondition</b>       | Managing staff needs to be logged in as a user  |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff will delete particular account of user which needs to be deleted.</li> <li>ii. Managing staff will delete the account.</li> <li>iii. End of user case</li> </ul> |
| <b>Alternative flow</b>   | N.A.  |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>Update user</b>   |
| <b>Brief Description:</b> | This allows the managing staff to update account of the user.  |
| <b>Actors</b>             | Managing staff   |
| <b>Precondition</b>       | Managing staff needs to be logged in as a user   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>iv. Managing staff will select particular account of user which needs to be updated.</li> <li>v. Managing staff will update the account.</li> <li>vi. End of user case</li> </ul> |
| <b>Alternative flow</b>   | <ul style="list-style-type: none"> <li>• System will display error message if the section are not filled and managing staff needs to enter missing information.</li> </ul>   |

|                           |   |
|---------------------------|---|
| <b>Use Case:</b>          | <b>View user list</b>   |
| <b>Brief Description:</b> | This allows the managing staff to view the list of users.   |
| <b>Actors</b>             | Managing staff  |
| <b>Precondition</b>       | Managing staff needs to be logged in as a user  |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff will view the list of list of users by clicking “View customer list ” or ‘view delivery person’ button.</li> <li>ii. Information get displayed in table.</li> <li>iii. End of user case</li> </ul> |
| <b>Alternative flow</b>   | N.A.  |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>Add Order</b>   |
| <b>Brief Description:</b> | This allows the managing staff to add or create order/delivery of their customer.  |
| <b>Actors</b>             | Managing staff   |
| <b>Precondition</b>       | Managing staff adds the order when customers approach for their delivery.  |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff will add the order of customer by filling in their details where they get the parcel id.</li> <li>ii. End of user case</li> </ul> |
| <b>Alternative flow</b>   | <ul style="list-style-type: none"> <li>• Details of order is not filled properly</li> </ul>  |

|                           |   |
|---------------------------|---|
| <b>Use Case:</b>          | <b>Update order</b>   |
| <b>Brief Description:</b> | This allows the managing staff to update the order of their customer.   |
| <b>Actors</b>             | Managing staff  |
| <b>Precondition</b>       | Managing staff updates the order that have been made.   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff search parcel by its parcel id of the parcel that need to be updated.</li> <li>ii. Managing staff will update the necessary information about the order.</li> <li>iii. End of user case</li> </ul> |
| <b>Alternative flow</b>   | <ul style="list-style-type: none"> <li>• Details of order are not filled properly and wrong number format.</li> </ul>   |

|                    |  |
|--------------------|--|
| Use Case:          | <b>Delete Order</b>  |
| Brief Description: | This allows the managing staff to delete the order of their customer.  |
| Actors             | Managing staff   |
| Precondition       | Managing staff needs to be logged in as the user for delete of order.  |
| Main Flow          | <ul style="list-style-type: none"> <li>i. Managing staff search parcel by its parcel id which need to be deleted.</li> <li>ii. End of user case</li> </ul> |
| Alternative flow   | N.A.   |

|                    |  |
|--------------------|--|
| Use Case:          | <b>View Order List</b>   |
| Brief Description: | This allows the managing staff and delivery staff to view the order list created by managing staff.  |
| Actors             | Managing staff and Delivery staff  |
| Precondition       | Either managing staff or delivery staff needs to be logged in as the user for viewing the list of order.   |
| Main Flow          | <ul style="list-style-type: none"> <li>i. Delivery staff and Managing staff will view the order approached by customer in table.</li> <li>ii. End of use case</li> </ul> |
| Alternative flow   | N.A.   |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>Update profile information</b>  |
| <b>Brief Description:</b> | This allows the delivery staff to update their own profile.  |
| <b>Actors</b>             | Delivery staff   |
| <b>Precondition</b>       | Delivery staff needs to login to update their profile.   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Delivery staff can view their profile and update if necessary by clicking “update button”.</li> <li>ii. End of use case</li> </ul> |
| <b>Alternative flow</b>   | If phone number is changed.  |

|                           |   |
|---------------------------|---|
| <b>Use Case:</b>          | <b>View Parcel Status</b>   |
| <b>Brief Description:</b> | This allows the customer to view the status of their parcel.  |
| <b>Actors</b>             | Customers   |
| <b>Precondition</b>       | Customer needs enter their parcel id as requested by the system.  |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. System will display their parcel status and can view current location of the parcel that they have ordered for delivery.</li> </ul> |
| <b>Alternative flow</b>   | N.A.  |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>Generate Invoice</b>  |
| <b>Brief Description:</b> | This allows the managing staff to generate the invoice.  |
| <b>Actors</b>             | Managing staff   |
| <b>Precondition</b>       | Managing staff needs to login as the user.   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Enter the information of parcel and add to the invoice generator table .</li> <li>ii. Select all the rows and click ‘generate invoice’ .</li> <li>iii. Invoice gets generated.</li> <li>iv. End of use case</li> </ul> |
| <b>Alternative flow</b>   | N.A.   |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>Add feedback</b>  |
| <b>Brief Description:</b> | This allows the managing staff to provide feedback about their work for delivery staff.                            |
| <b>Actors</b>             | Managing staff   |
| <b>Precondition</b>       | Managing staff needs to login as the user.   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Managing staff will add feedback</li> <li>ii. End of use case</li> </ul> |
| <b>Alternative flow</b>   | N.A.   |

|                           |  |
|---------------------------|--|
| <b>Use Case:</b>          | <b>View feedback</b>   |
| <b>Brief Description:</b> | This allows the delivery staff to view feedback which is added by managing staff.  |
| <b>Actors</b>             | Delivery staff   |
| <b>Precondition</b>       | Delivery staff needs to login as the user.   |
| <b>Main Flow</b>          | <ul style="list-style-type: none"> <li>i. Delivery staff will view feedbacks given by managing staff which notify them about their work quality.</li> <li>ii. End of use case</li> </ul> |
| <b>Alternative flow</b>   | N.A.   |

### 3. OOP and System Implementation

OOP is a programming paradigm which relies on the concept of class and objects. A data field with unique attributes and behaviour is referred to as an object. This approach of programming is ideal for large, sophisticated and frequently updated or maintained programs (Gills & Lewis, 2021). The principles of OOP that have been applied in our assessment are described below;

#### 3.1. Class and Object

A class is a blueprint or prototype defined by the user which is used to create objects. It also provides initial state values (variables) and action implementation (function or methods). While an object is defined as an abstract data type created by a developer. Objects are also known as classes in most programming languages.

##### Class

```
/*
public class User {

    private String username;
    private String passWord;
```

##### Object

```
User user = new User(usernamekb,passwordkb)
user.setUsername(usernamekb);
user.setPassword(passwordkb);
user.login();
jTextField_un.setText(null);
jPasswordField_pass.setText(null);
```

## 3.2. Encapsulations

Encapsulation is one of the main principles of OOP. It is the mechanism that connects data to the code it works with. It is a protective shield that stops code from accessing data outside of the shield. The user won't have any idea of the class's internal implementation which can be also called as data or information hiding ("Encapsulation in Java - GeeksforGeeks", 2021). It also promotes re-usability and makes it simple to adapt to new requirements. For achieving encapsulation in Java, we must declare a class's variables as private and provide public getter and setter methods to change and inspect the variables values.

```
public class User {  
  
    private String username;  
    private String passWord;  
  
    public User(String username, String passWord) {  
        this.username = username;  
        this.passWord = passWord;  
    }  
    public String getUsername() {  
        return username;  
    }  
  
    public void setUsername(String username) {  
        this.username = username;  
    }  
  
    public String getPassword() {  
        return passWord;  
    }  
  
    public void setPassword(String passWord) {  
        this.passWord = passWord;  
    }  
}
```

### 3.3. Inheritance

Inheritance is an important principle of OOP. In java, it is the mechanism that allows one class to inherit the features of another class. It is used to define a variety of exceptions, add custom logic to existing framework, and link our domain model to a database.

Hierarchical inheritance is chosen in between different types of inheritance because a base class is inherited from more than one class ("Inheritance in Java - GeeksforGeeks", 2021). The child class inherits all of the parent class's common features and vice versa according to Syntax. Thus, it is feasible to inherit other function rather than only common features. The extension keyword "extends" is used to define the class that subclass extends.

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

    /**
     */
    public Invoice() {
        initComponents();
    }

    public Invoice(String name, String date, String idd, String total, String pay, String balance) {
        jLabel_customerName.setText(name);
        jLabel_date.setText(date);
        jLabel_parcelId.setText(idd);
        jLabel_total.setText(total);
        jLabel_pay.setText(pay);
        jLabel_balance.setText(balance);
    }
}
```

### 3.4. Constructor

A constructor in a java is a method for initialising objects. It is mainly used to set primary values for object attributes. Every class has constructor because Java includes a default constructor which sets all members variables to 0 by default. The default constructor will be useless if we define our own constructor ("Java - Constructors", 2021).

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

    /**
     */
    public Invoice() {
        initComponents();
    }

    public Invoice(String name, String date, String idd, String total, String pay, String balance) {
        jLabel_customerName.setText(name);
        jLabel_date.setText(date);
        jLabel_parcelId.setText(idd);
        jLabel_total.setText(total);
        jLabel_pay.setText(pay);
        jLabel_balance.setText(balance);
    }
}
```

### 3.5 Polymorphism

Polymorphism in java can be defined as a concept in which user can perform single action in several ways. Static and dynamic polymorphisms are two types of polymorphism where java supports. Static polymorphism permits us to implement different methods within the same class using same name but distinct set of parameters. It is assumed that the parameters would be in different orders ("Polymorphism in Java - javatpoint", 2021).

## 4. User Manual

### 4.1. Home page

This is the very first page of this application. There are three buttons available in this page that is customer, managing staff and delivery person along with exit button. If ‘**customer**’ button is pressed then system asked us to enter the parcel id which helps us to track the location of the parcel. Similarly, when ‘**managing staff**’ and ‘**delivery staff**’ buttons are clicked then system asked us to login . With out login no users are able to perform the activity. The home page of the application is shown below:

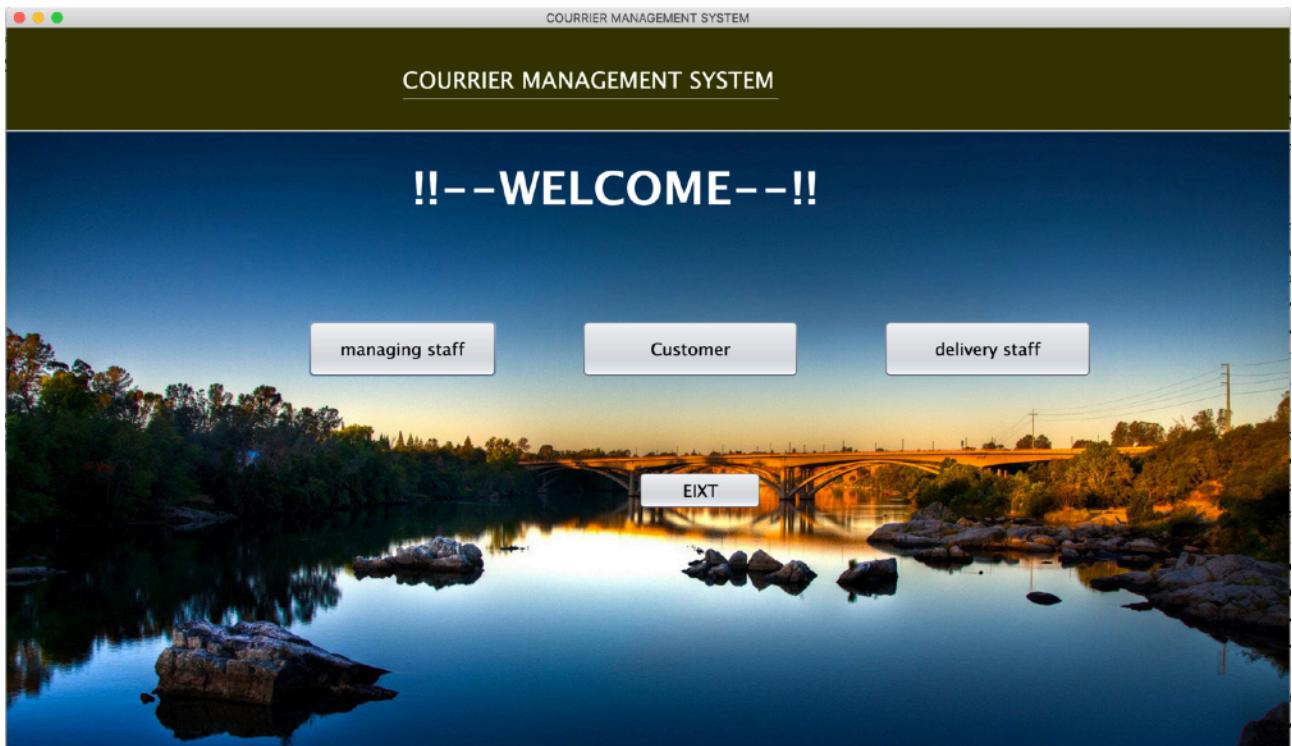


Figure 1: Home page

## 4.2. Features

The activities that managing staff can perform and need to do in order to use their power are described below with their interface design:

### 4.2.1. Login form

This is the login form of managing staff. So, in order to use the facility provided to the managing staff and delivery staff they need to login to the system through this form. They need to enter username and password as registered before. But If the staff is new then he/she need to register into the system for that they need to click the new register button .

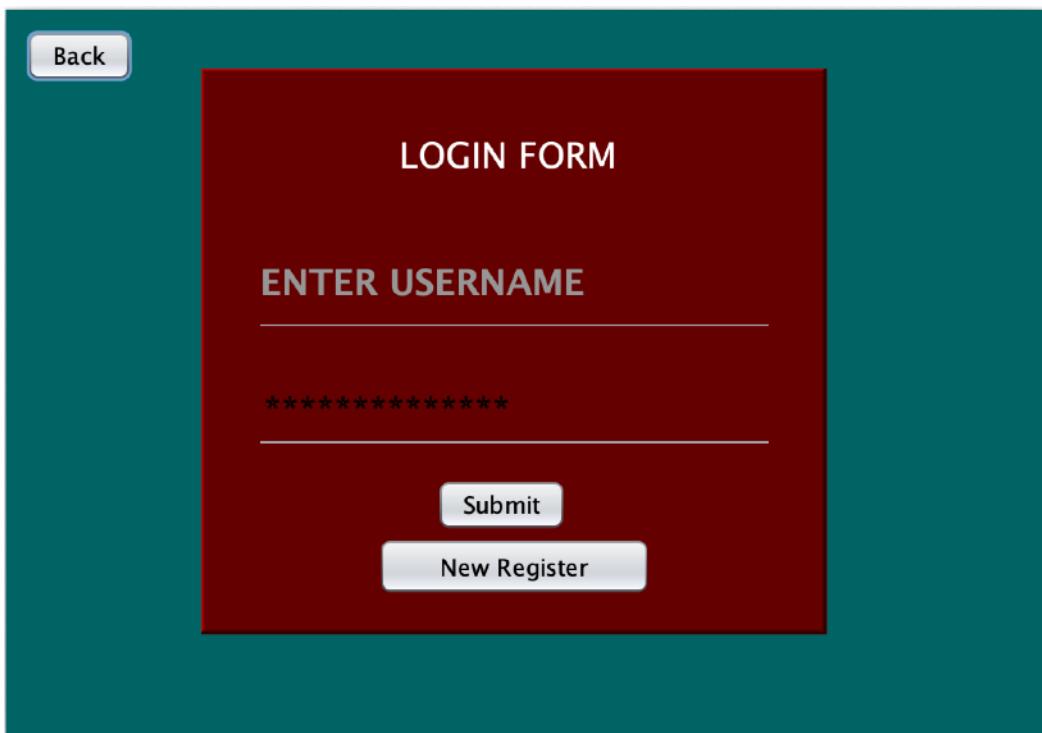


Figure 2: Managing Staff Login Form

#### 4.2.2. Register Form

As soon as new register button is clicked then registration form appears where they need to enter the information as requested . They need to provide unique username.If the registration is completed then they can simply click “sign up “ to enter into to login form. Here they can enter the username and password as registered before in order to preform their task. The registered data are stores in txt file

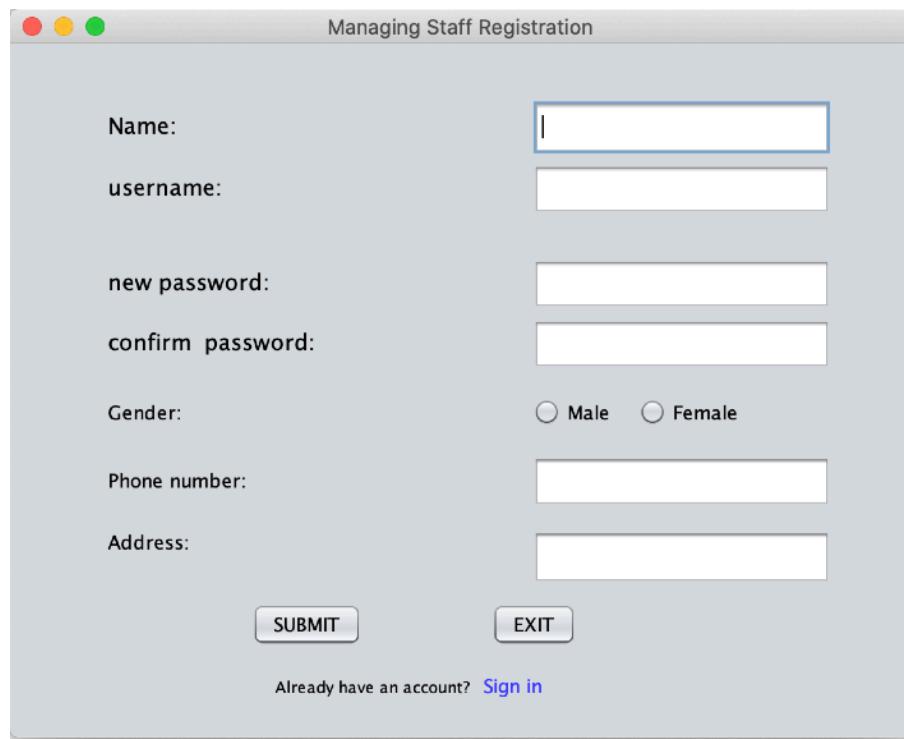


Figure 3: Managing Staff Registration Form

#### 4.3. Managing Staff Panel

The user enter into the managing staff panel as shown below after login is successful. The features that staff can perform are demonstrated at the left side of the frame in the form of buttons. The features are user account management , order management, Manage report. Similarly, they can simply logout of the system by clicking logout button.

### 4.3.1. User Management

- User account Management

As soon as user click the user account management button on Managing staff panel they can view “customer management” and delivery account management” buttons as shown on following user interface design:

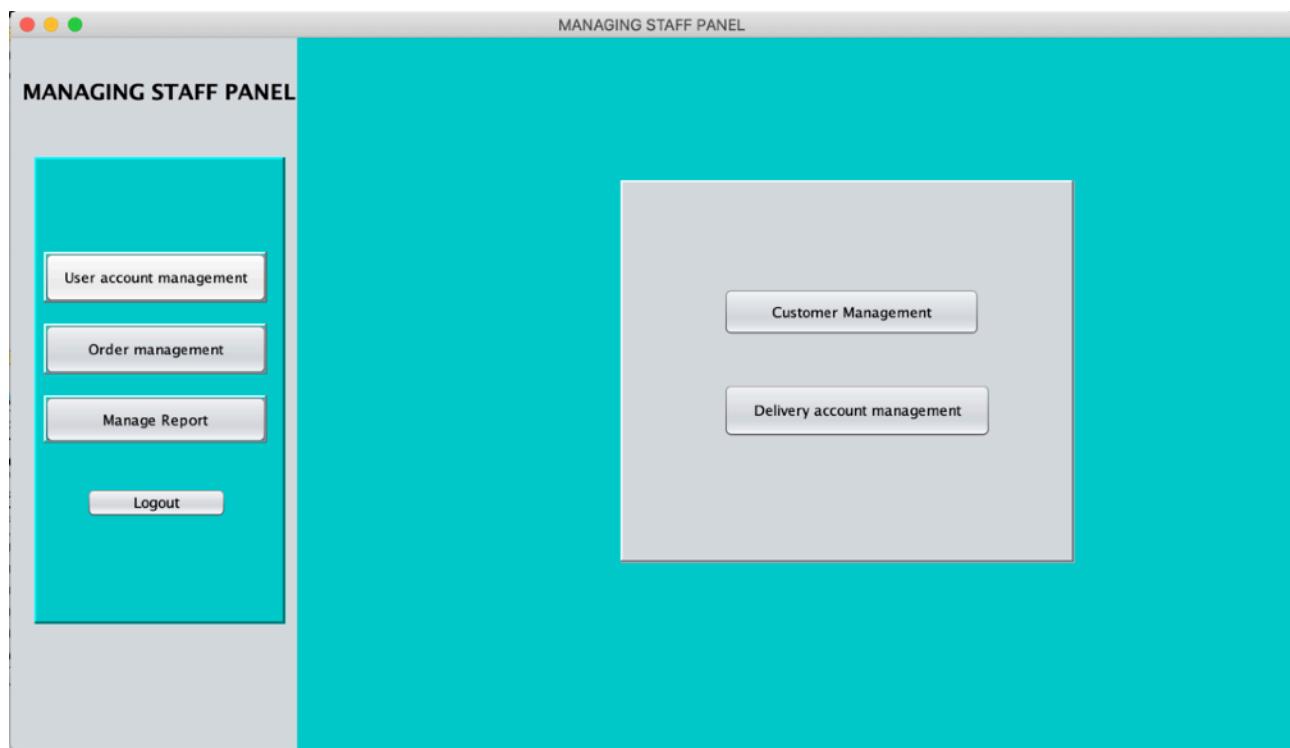


Figure 4: User Account Management

Customer management includes the features like register new customer , search customer account, update customer account, view customers list and delete customer account. So, while registering customer each customer accounts are assigned by unique id , with the help of that id managing staff can search customer account which helps to save time as well as manpower . Even if the admin wants to delete or update a customer account he/she can simply enter unique id assigned to each customer account . With the help of this id managing staff can perform all this activity within a second. Exit button helps user to return to the previous frame. The interface design of the customer management panel along with the features is shown below:

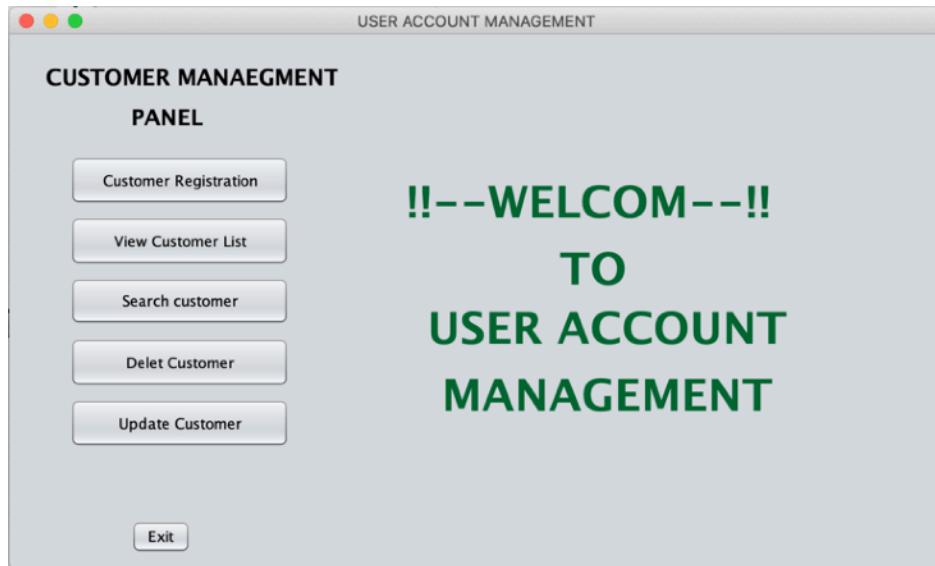


Figure 5: Customer Account Management

Similarly, if staff clicks the delivery account management button then he/she enters into delivery account management panel where they can add delivery person, view delivery persons list, search delivery person, delete person and update their information. In addition to this, each delivery person is assigned with unique id which helps admin to identify each account individually. As shown on the figure admin need to enter unique id on the search bar and click search button. As soon as search button is pressed then the information of the account with that id are displayed in each respective text field. If user wants to delete the account he/she can simply click delete button where as click update button if wanted to update the information.

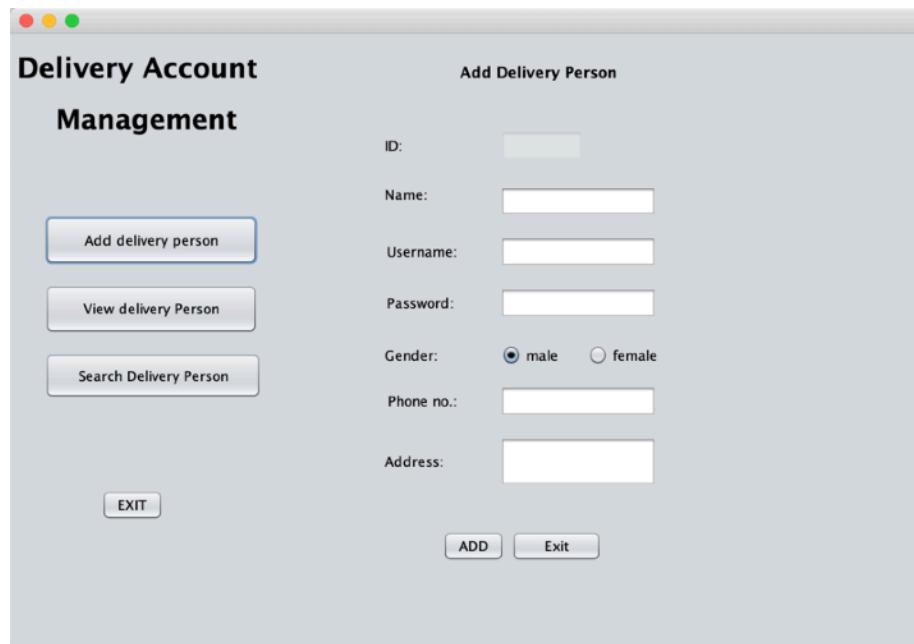


Figure 6: Delivery Account Management

| ID  | Name    | username  | password | gender | Phone no.  | Address |
|-----|---------|-----------|----------|--------|------------|---------|
| 231 | suraj   | suraj     | suraj    | male   | 3948394    | pokhara |
| 232 | sandesh | sandsh123 | sandesh  | male   | 9846779... | Pokhara |
| 235 | Kishor  | kishor123 | Kishor   | male   | 098807     | pokhara |
| 236 | Nabin   | Chhetri   | Nabin123 | male   | 38475935   | sdgdsg  |

Figure 7: View Delivery Person

|                                       |   |
|---------------------------------------|---|
| 232                                   | <input type="button" value="SEARCH"/>         |
| ID:                                   | 232   |
| Name:                                 | sandesh                                       |
| Username:                             | sandsh123                                     |
| Password:                             | *****   |
| Gender:                               | male  |
| Phone no.:                            | 9846779123                                    |
| Address:                              | kathmandu                                     |
| <input type="button" value="Update"/> | <input type="button" value="Delete account"/> |

Figure 8: Search Delivery Account

### **4.3.2. Order Management**

When staff clicks the order management option then a new panel gets visible on managing staff panel. In this panel buttons like add delivery, view delivery along with exit button are visible. ‘Exit’ button simply brings us back to the “Managing Staff Panel”. As soon as staff clicks “add delivery” button, a frame is appeared where they can add all the detail information of the order. Each delivery order will also be assigned by unique id called parcel id which uniquely identify each parcel registered into the system. After the textfields are filled with correct information and add button is pressed then the data get registered and stored on txt file , Also some of the data required for invoice are added on the table on the right side of the frame. If the customer has got delivery order with two different items like clothes and document the information need to be separately registered but the invoice can be same. If the rows of the table are selected and the button called “generate invoice” is clicked then the invoice or bill of that order gets appeared.

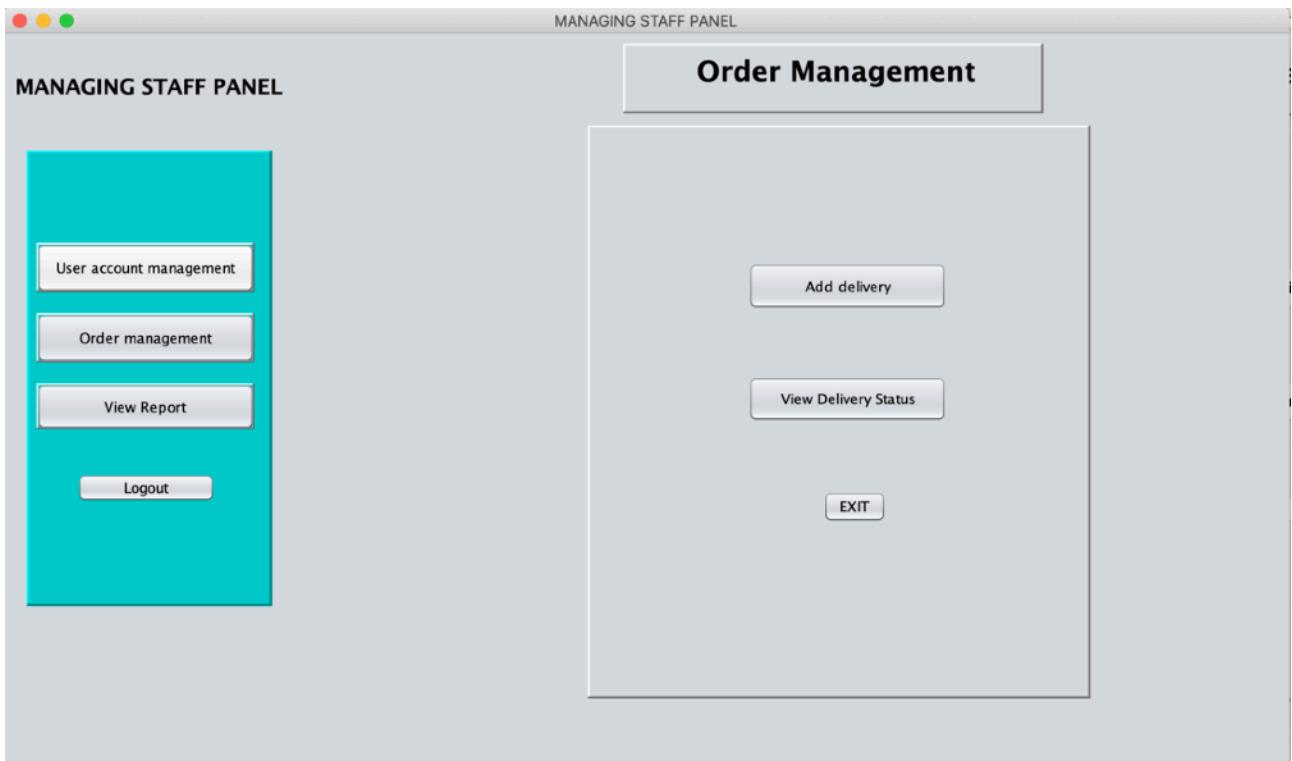


Figure 9: Order Management

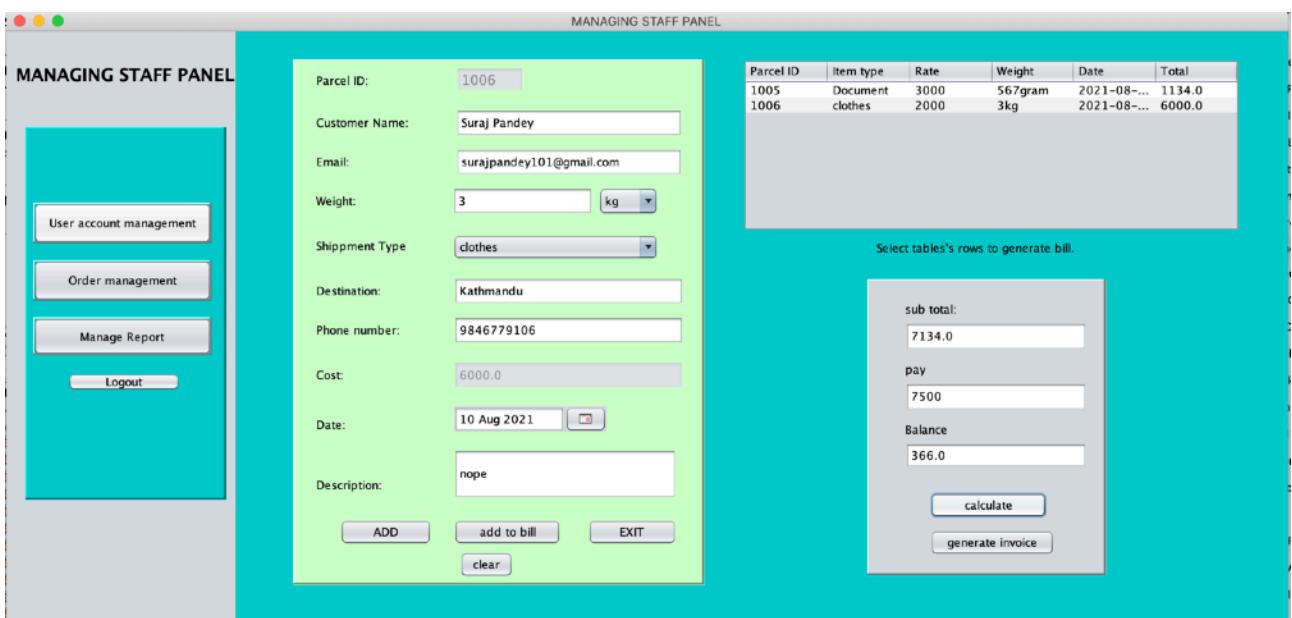


Figure 10: Add Delivery

# INVOICE

**Info courier** Info@courrier.com  
Pokhara, Gandaki, Nepal +977 061464101  
33700

Invoice no.: 1007

Bill to: Suraj Pandey

Date: 2021-08-10

| Parcel ID | Item type | Rate | Weight  | Date       | Total  |
|-----------|-----------|------|---------|------------|--------|
| 1005      | Document  | 3000 | 567gram | 2021-08-10 | 1134.0 |
| 1006      | clothes   | 2000 | 3kg     | 2021-08-10 | 6000.0 |

Note: To check the delivery status please check  
on [www.infocourier.com](http://www.infocourier.com) using your parcel id.

Sub total: 7134.0  
pay 7500  
Balance: 366.0

Save

exit

Figure 11: Invoice

## View Delivery Status

When we click the view delivery status then the detail information of the parcels that are registered into the system till now will appear on a table. Similarly, can delete the delivery by entering parcel id on search bar and clicking delete button and update by clicking on update button through which staff can view the current location and parcel status whether the parcel is delivered or not as shown on the figure below:

**Delivery Status**

| Parcel ID | Customer Name | Customer Email | Weight   | Shipment Type | Destination | Phone number | cost       | Date       | Description | Current Location | Status   |
|-----------|---------------|----------------|----------|---------------|-------------|--------------|------------|------------|-------------|------------------|----------|
| 1001      | Suraj         | suraj101@...   | 123gram  | Document      | pokhara     | 93875395     | Rs.246.0   | 2021-08-08 | nothing     | delivered        | approved |
| 1002      | sandesh       | sandesh        | 2kg      | clothes       | Biratnagar  | 39847593...  | Rs.4000.0  | 2021-08-08 | no          | Bharatpur        | approved |
| 1003      | Suraj         | surajpandey    | 12kg     | clothes       | Kathmandu   | 9283420      | Rs.24000.0 | 2021-08-08 | --          | Mugling          | approved |
| 1007      | SurajPandey   | surajpand...   | 4kg      | clothes       | Biratnagar  | 845698376    | Rs.8000.0  | 2021-08-08 | --          | Birgunj          | approved |
| 1008      | Sandesh Giri  | sandy12@...    | 345gram  | clothes       | Australia   | 9845793      | Rs.690.0   | 2021-08-08 | --          | Thailand         | approved |
| 1009      | Ram           | ram@gmail...   | 460gram  | Document      | Ayodhya     | 89345937     | Rs.920.0   | 2021-08-08 | --          | Birgunj          | approved |
| 1010      | shyam         | shyame         | 2kg      | clothes       | Butwal      | 9798         | Rs.4000.0  | 2021-08-08 | nothing     | Narayan G...     | approved |
| 1011      | shyam         | shyame         | 4kg      | clothes       | Baglung     | 9798453      | Rs.8000.0  | 2021-08-08 | --          | Kushma           | approved |
| 1013      | John          | John@gmai...   | 1275gram | clothes       | U.K.        | 983647634    | Rs.2550.0  | 2021-08-08 | --          | Delhi            | approved |

**Parcel ID:** 1008      **Search**

**Customer name:** Sandesh Giri      **cost:** Rs.690.0      **update**

**Email:** sandy12@gmail.com      **Date:** 2021-08-09

**Weight:** 345gram      **Description:** --      **delete**

**Shippmetn type:** clothes      **current location:** Thailand

**Destination:** Australia      **status:** approved

Figure 12: View Delivery Status

#### 4.3.3. Report Management

In report management managing staff can view the invoice report which include invoice id, customer name, transaction date and total amount. The report of invoice can be displayed as below:

**Invoice Report**

| Invoice ID | Customer Name | Transaction Date | Total Amount |
|------------|---------------|------------------|--------------|
| 100002     | SurajPandey   | 2021-08-08       | 9134.0       |
| 100003     | Sandesh Giri  | 2021-08-09       | 690.0        |
| 100004     | Ram           | 2021-08-09       | 920.0        |
| 100005     | shyam         | 2021-08-09       | 12920.0      |
| 100006     | Hari          | 2021-08-10       | 24000.0      |
| 100007     | John          | 2021-08-10       | 26550.0      |

Figure 13: View Report

#### 4.3.4. Feedback management

Managing staff are also able to send feedback to the delivery staff. The UI is shown below:

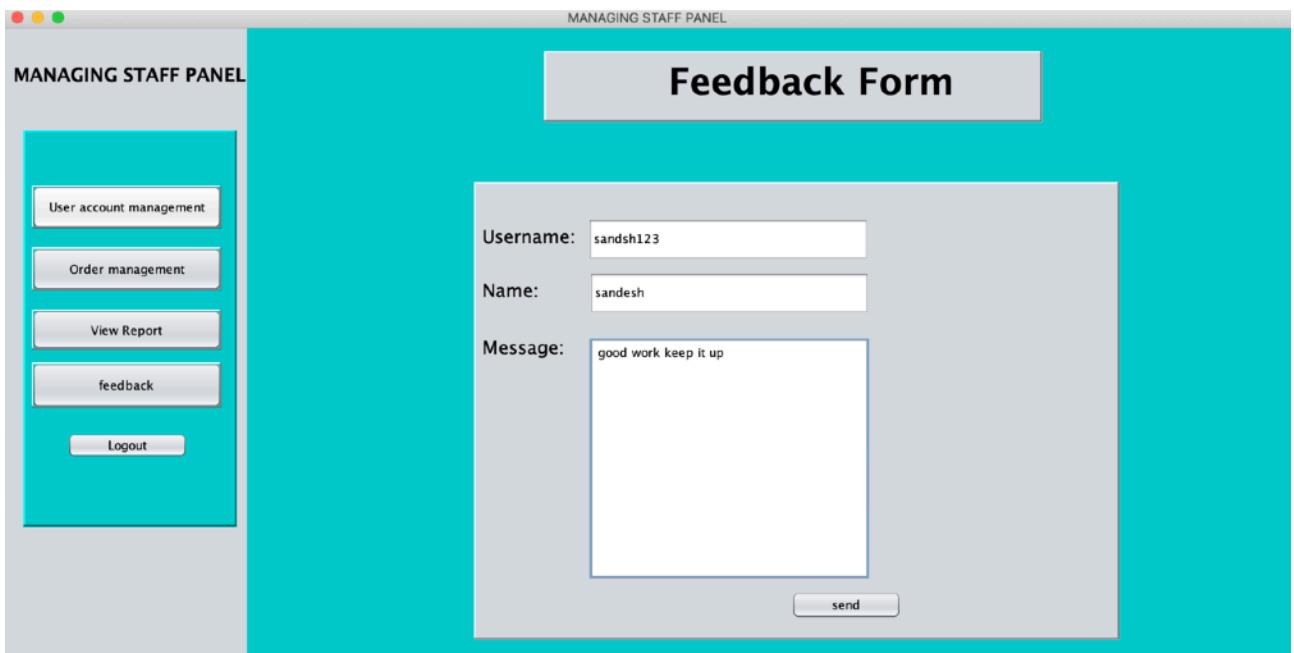


Figure 14: Feedback management

#### 4.4. Customer features

As we know that each delivery order is assigned with unique id called parcel id , customer can view the current location of their parcel by entering id into the search bar as below:

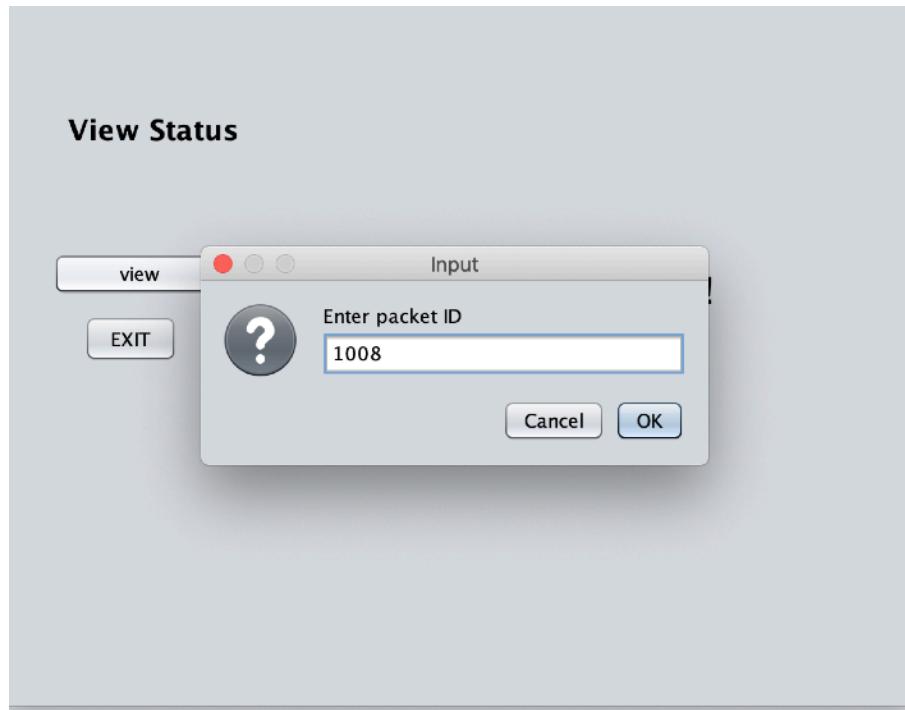


Figure 15: Track Parcel

A screenshot of a computer application window titled "View Status". The window displays various details about a parcel, organized in a grid-like form. The details are as follows:

|                  |                   |
|------------------|-------------------|
| Customer name:   | Sandesh Giri      |
| Email:           | sandy12@gmail.com |
| Weight:          | 345gram           |
| Type:            | clothes           |
| Destination:     | Australia         |
| Phone number:    | 9845793           |
| Cost:            | Rs.690.0          |
| Date:            | 2021-08-09        |
| Current Location | Thailand          |
| status:          | approved          |

To the left of the main window, there is a vertical toolbar with a "view" button and an "EXIT" button.

Figure 16: View Current Location

## 4.5. Delivery Staff Panel

### 4.5.1. Login Form

Before using the power given to the delivery staff they also need to login into the system . Delivery person cannot register themselves rather they are registered by managing staff as mentioned above. They can login with the help of username and password provided to them by managing staff. The login form of the delivery staff is shown below:



Figure 17: Delivery Staff Login Form

### 4.5.2. View Order

After being login into the system they can perform activities like view order, update delivery status and view individual profile. After clicking the view Profile button they can view their profile and can update if needed. They can only update current location of the parcel. To update location they need to select the row of the table, make changes and click the button called “update location” as shown in the figure below:

A screenshot of a 'Order List' window. At the top is a table titled 'Order List' with columns: ID, Customer name, Email, weight, and Type. The table contains several rows of data. On the left side of the main window are three buttons: 'view delivery', 'view Profile', and 'EXIT'. On the right side, there is a detailed view of a selected row from the table. This view includes fields for Delivery ID (1005), Customer Name (Suraj Pandey), Email (surajpandey101@gmail.com), Weight (567gram), Type (Document), Destination (Kathmandu), Phone no.: (9846779105), cost: (Rs.1134.0), Date: (2021-08-10), Description: (none), and current Location (Damauli). A message 'data updated successfully' is displayed above the 'update location' button. There are also 'update location' and 'Exit' buttons at the bottom.

Figure 18: View Order

#### 4.5.3. View and Update Profile

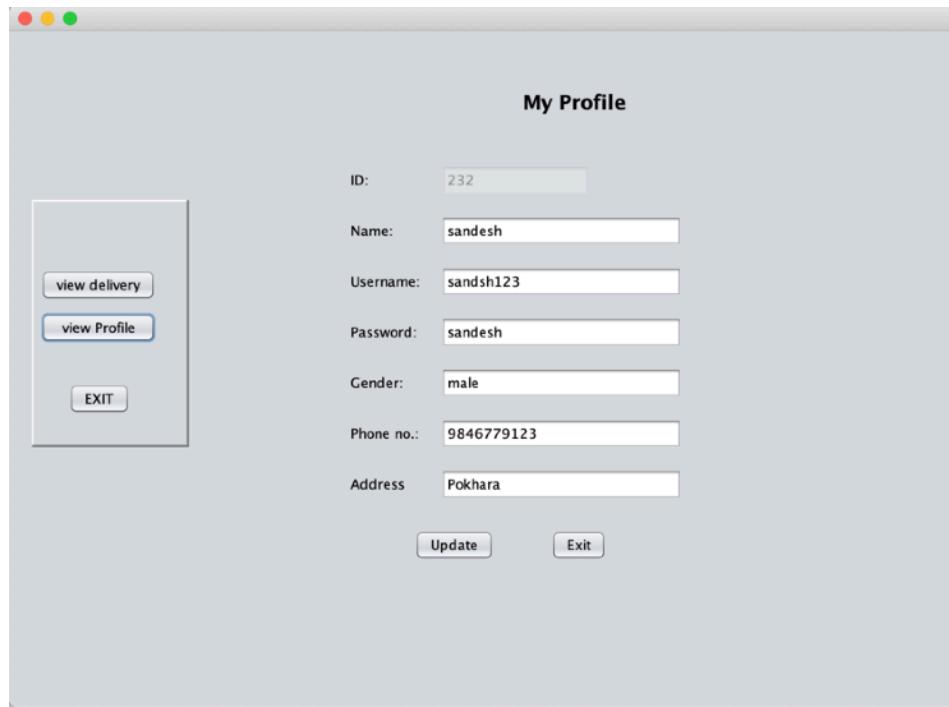


Figure 19: View Individual Profile

#### 4.5.4. View Feedback

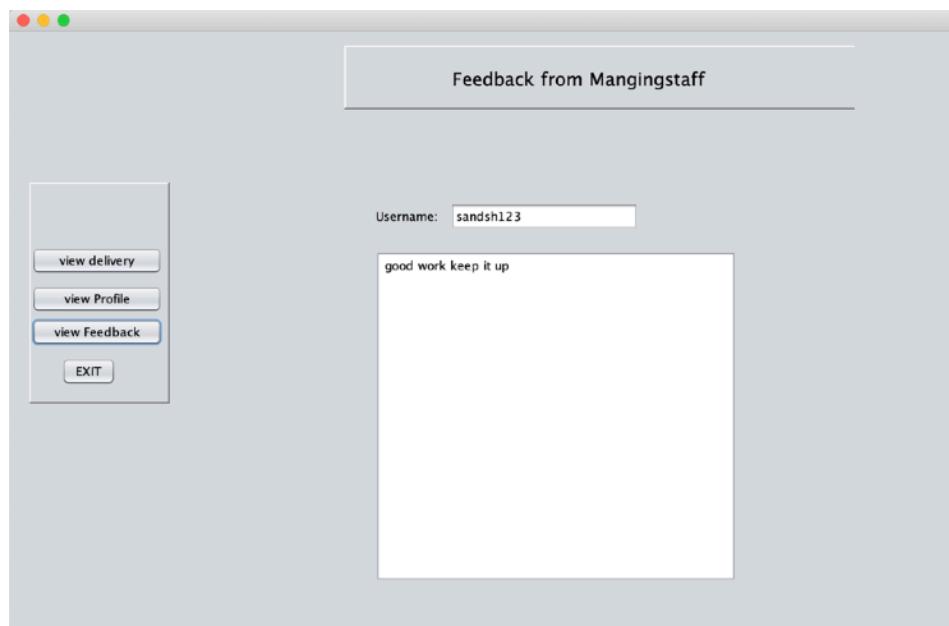


Figure 20: View Feedback

## 5. Conclusion

This project was developed under java programming language following object oriented programming concept like class and object creation, encapsulation, inheritance, polymorphism and use of constructor. This system allows us to handle users account, manage order, manage report, create invoice, view status, update status and so on. Most of the features are available to the managing staff whereas delivery staff can only view order list, update status, view individual profile and update information if necessary. And they can also view the feedback provided by managing staff.

## 6. References

- ALTVATER, A. (2020). What Are OOP Concepts in Java? The Four Main OOP Concepts in Java, How They Work, Examples, and More. Retrieved 8 August 2021, from <https://stackify.com/oops-concepts-in-java/>
- Capka, D. (2021). Lesson 7 - Inheritance and polymorphism in Java. Retrieved 10 August 2021, from <https://www.ictdemy.com/java/oop/inheritance-and-polymorphism-in-java>
- Encapsulation in Java - GeeksforGeeks. (2021). Retrieved 9 August 2021, from <https://www.geeksforgeeks.org/encapsulation-in-java/>
- Gills, A., & Lewis, S. (2021). What is Object-Oriented Programming (OOP)? Retrieved 8 August 2021, from <https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP>
- Inheritance in Java - GeeksforGeeks. (2021). Retrieved 7 August 2021, from <https://www.geeksforgeeks.org/inheritance-in-java/>
- Java - Constructors. (2021). Retrieved 11 August 2021, from [https://www.tutorialspoint.com/java/java\\_constructors.htm](https://www.tutorialspoint.com/java/java_constructors.htm)
- Polymorphism in Java - javatpoint. (2021). Retrieved 10 August 2021, from <https://www.javatpoint.com/runtime-polymorphism-in-java>

