# Table of Contents

[1 Project Introduction 1](#_Toc80078274)

[2 Existing System Overview 2](#_Toc80078275)

[2.1 Statement of Problem 3](#_Toc80078276)

[2.2 Objective 3](#_Toc80078277)

[2.3 Scope 3](#_Toc80078278)

[2.4 Limitation 3](#_Toc80078279)

[3 Research Methodology 4](#_Toc80078280)

[3.1 Feasibility Analysis 4](#_Toc80078281)

[3.2 Economic Feasibility 4](#_Toc80078282)

[3.3 Operational Feasibility 4](#_Toc80078283)

[3.4 Technical Feasibility 4](#_Toc80078284)

[3.5 Schedule Feasibility 5](#_Toc80078285)

[4 Software Development Methodology 6](#_Toc80078286)

[4.1 Functional and Non-Functional Requirement Specification 6](#_Toc80078287)

[4.1.1 Functional Requirement 6](#_Toc80078288)

[4.1.2 Non-Functional Requirement 7](#_Toc80078289)

[5 Software Development 8](#_Toc80078290)

[5.1 Use Case Diagram 8](#_Toc80078291)

[5.2 Class Diagram 9](#_Toc80078292)

[5.3 Activity Diagram 10](#_Toc80078293)

[5.4 GANTT chart 11](#_Toc80078294)

[6 Hardware and Software Requirement 13](#_Toc80078295)

[7 Backup and Recovery 14](#_Toc80078296)

[8 Design and Implementation 15](#_Toc80078297)

[8.1 Class 15](#_Toc80078298)

[8.2 Object 15](#_Toc80078299)

[8.3 Encapsulation 15](#_Toc80078300)

[8.4 Constructor 16](#_Toc80078301)

[8.5 Inheritance 17](#_Toc80078302)

[8.6 Interface 17](#_Toc80078303)

[8.7 Polymorphism 18](#_Toc80078304)

[9 Screenshot 19](#_Toc80078305)

[10 Testing Strategies 32](#_Toc80078306)

[10.1 Unit Testing: 32](#_Toc80078307)

[10.2 System Testing: 32](#_Toc80078308)

[10.3 Performance Testing: 32](#_Toc80078309)

[11 Conclusions 34](#_Toc80078310)

[12 References 35](#_Toc80078311)

# Table of Figures

[Figure 1: Use Case Diagram 8](#_Toc80078166)

[Figure 2: Class Diagram 9](#_Toc80078167)

[Figure 3: Activity Diagram 10](#_Toc80078168)

[Figure 4: Gantt chart Timeline 11](#_Toc80078169)

[Figure 5: Gantt chart Timeline 2 12](#_Toc80078170)

[Figure 6: Class 15](#_Toc80078171)

[Figure 7: Object 15](#_Toc80078172)

[Figure 8: Encapsulation 16](#_Toc80078173)

[Figure 9: Constructor 16](#_Toc80078174)

[Figure 10: Inheritance 17](#_Toc80078175)

[Figure 11: Interface 17](#_Toc80078176)

[Figure 12: Implementation 17](file:///C:\Users\dpak1\OneDrive\Old\Desktop\JAVA%20BURN\NP000390-NP000410-CT038-3-2-OODJ.docx#_Toc80078177)

[Figure 13: Polymorphism 18](#_Toc80078178)

[Figure 14: User Type 19](#_Toc80078179)

[Figure 15: Login for Admin 20](file:///C:\Users\dpak1\OneDrive\Old\Desktop\JAVA%20BURN\NP000390-NP000410-CT038-3-2-OODJ.docx#_Toc80078180)

[Figure 16: Login for Employee 20](file:///C:\Users\dpak1\OneDrive\Old\Desktop\JAVA%20BURN\NP000390-NP000410-CT038-3-2-OODJ.docx#_Toc80078181)

[Figure 17: Admin Dashboard 21](#_Toc80078182)

[Figure 18: Employee Dashboard 21](#_Toc80078183)

[Figure 19: Courier Details Form 22](#_Toc80078184)

[Figure 20: Customer Registration Form 23](#_Toc80078185)

[Figure 21: Employee Registration Form 24](#_Toc80078186)

[Figure 22: Feedback 25](#_Toc80078187)

[Figure 23: Manage Courier 26](#_Toc80078188)

[Figure 24: Manage Customer 26](#_Toc80078189)

[Figure 25: Manage Employee 27](#_Toc80078190)

[Figure 26: Update Courier 27](#_Toc80078191)

[Figure 27: Update Customer 28](#_Toc80078192)

[Figure 28: Update Employee 28](#_Toc80078193)

[Figure 29: Employee Details 29](#_Toc80078194)

[Figure 30: Courier Details 29](#_Toc80078195)

[Figure 31: Customer Details 30](#_Toc80078196)

[Figure 32: Report 30](#_Toc80078197)

[Figure 33: Feedback 31](#_Toc80078198)

# Project Introduction

Online Courier System is the system which enables the people to send and receive parcels or documents from one place to another through the use of a computerized system. This project is all about developing the courier service system for sending, delivering and receiving the couriers. Here we have developed the online system where we allow customers to send and receive the courier item through the use of the internet. Similarly, the admin can do everything on the system and delivery staff look after the customer orders and deliver them. This system runs through the use of the internet.

This online system is developed focusing mainly on two types of end user i.e. Managing Staff and Delivery Staff.

**Managing Staff**

In this system managing staff refers to the administrator who holds the authority to handle almost everything possible inside the system so there are the most main and powerful users. They have the authority to add, view, modify, remove or delete the customer, employees and orders. Only the admin can provide access for registration of the new user to the system. Admin is the one who can observe all the activity and generate the delivery report.

**Delivery Staff**

In this system delivery Staff are the ones who are given responsibility to handle the deliveries. They are given permission to make changes or update their own profile. They are responsible for delivery of the package who ensure that it gets to the right customer within the time period. They can view the orders and few details of the customer to deliver the product to the exact customer.

# Existing System Overview

As a developer, it is critical to the current setups done by the company to run their business or any organization. A developer must study, analyze and understand the existing problem to find out the suitable solution of the problem. In the current scenario of the Courier Service System all the organization's works have been carried out traditionally i.e. on papers. This has made the operation of the organization slow and inefficient. Taking in consideration to solve this problem we will be updating the entire organization's operation from paper documentations to online-software base systems. After introducing the online Courier Service System, more things can be done in less time which will increase the efficiency of the organization. Both the managing staff and the delivery staff will have access to the data and information as per their roles which will fast track the work and the organization entirely.

## Statement of Problem

The existing Courier Service System is working solely based on physical documents and few excel files which are not integrated with each other. The data and information are not circulated with all the staff on time as the system is not real-time. The way the organization is documenting the data, it is making the process slow where efficiency and speed is compromised.

## Objective

The digitalization of the Courier Service System from conventional too well organized format.

1. To enable managing staff and delivery staff to handle account, feedback, report management and delivery, individual profile management respectively.
2. To make the system secure and reliable
3. To track the Courier while processing in Courier Service System
4. To reduce data redundancy and time consumption

## Scope

1. To validate the staff login for security purposes.
2. It helps in deciding the roles of the staff and give them designated permissions
3. To ensure the accuracy of the report on the Courier Service System

## Limitation

1. No use of database.
2. Expensive to maintain.

# Research Methodology

## Feasibility Analysis

A feasibility analysis evaluates the whether the proposed system is feasible. It also studied software products in terms of how useful product development will be for business. It is performed in order to determine whether a software product is fit for development, implementation, and project value to the business, among other things. (GeeksforGeek, 2020)

## Economic Feasibility

Economic feasibility involves the study of both cost and benefit of the project. In this study all the development cost for the project is determined. All the cost for hardware, software, manpower, development, operational costs are analyzed to find out whether the project is a good financial decisions or not for the organization.

## Operational Feasibility

In operational feasibility degree of service to requirements is evaluated as well as study whether the developed product will operate properly. The operational cost is determined by the usability of the product and also suggestions gathered by the team of software developers for operational feasibility.

## Technical Feasibility

In technical feasibility evaluation and examination of current resources of both hardware and software, as well as required technologies is made in order to develop a project. This determines whether available resources and technologies are being used for the project.

## Schedule Feasibility

Schedule feasibility study measure the degree to which a deadline for a strategy, plan, project, or process is practical and possible. It is conducted in order to determine if the project can be completed within estimated time frame or not.

# Software Development Methodology

Extreme Programming (XP) is one of the software development methods that improve software quality and make it more responsive to customer requirements and also follows agile method principles. After studying the requirement of the assignment XP method was chosen as we found it to be the most suitable one for this project on the Online Courier Service System. We found the XP method that meets our project needs as the developed system is simple interfaced but high quality software. XP methods follow all the agile principles as well as the dynamic customer venture for the development cycle, group collaboration, and iterative improvement cycles. This supports teamwork and makes working of software faster.

Benefits of Extreme Programming (XP):

1. Saving cost and time is one of the greatest benefits of XP as it eliminates all the unproductive activities.
2. It reduces any risks to programming and failures of the projects as various testing is conducted through the development session.
3. It provides constant feedback through which change can be made easily whenever needed.

## Functional and Non-Functional Requirement Specification

### Functional Requirement

1. Managing staff/Admin and Delivery staff have access to the system.
2. Dashboard contains many more options.
3. Managing staff have authority to add, view, modify and delete the customer, employee and orders.
4. Delivery staff can view orders and feedbacks.
5. Delivery staff can manage their profile on their own.

### Non-Functional Requirement

1. Performance Requirement is ensure by the efficiency of the program.
2. Software Quality is top notch as it fulfills the basic requirement of the end users.
3. Security Requirement is considered as top priority and OOPs concept is used to ensure the security.

# Software Development

## Use Case Diagram

Use case Diagram refers to such diagram that capture the dynamic aspect of a system or subsystem. It contains the components like actors, use case and their relationships. There are two actors who are the end users of the program.

(tutorialspoint, 2021)

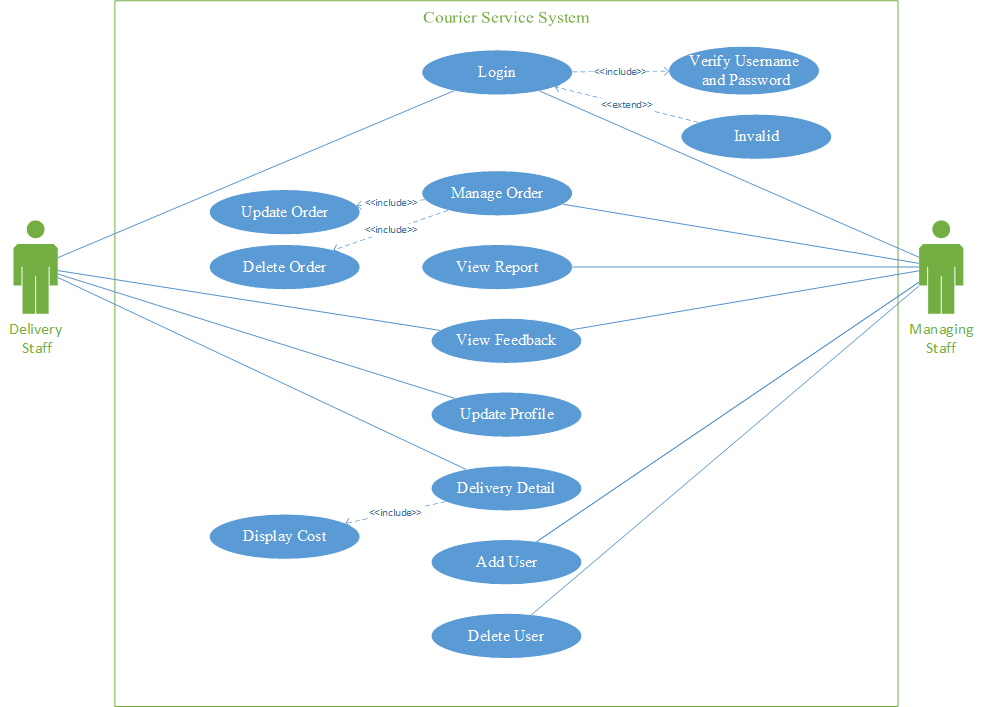


Figure 1: Use Case Diagram

## Class Diagram

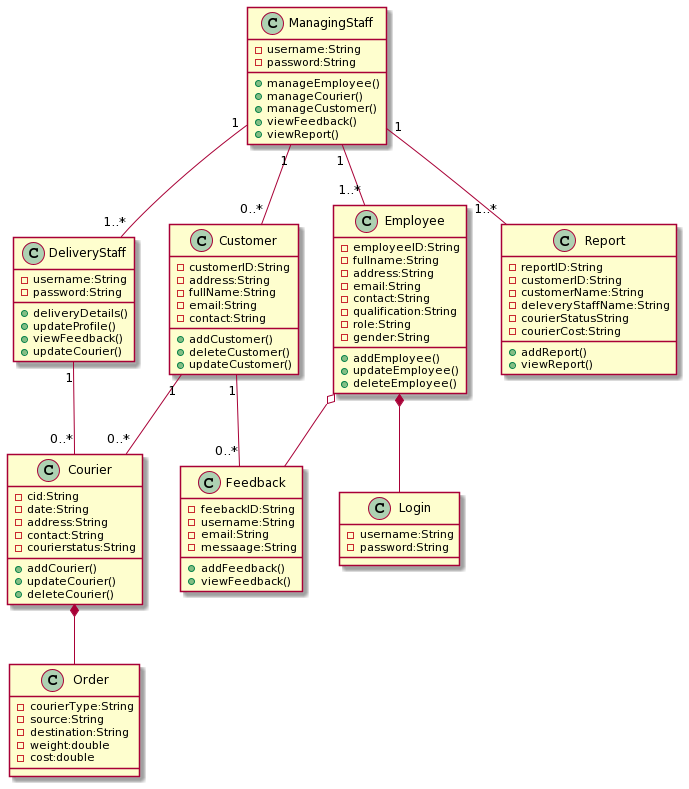
Class Diagram is a diagram that describe or define classes and their and relationships while designing and developing a software. This diagram basically represent the classes’ names and attributes, connections between the classes and methods too. (Java-Programming, 2021) In Figure 2 entities and relationships is shown with the existing classes of Courier Service System.

Figure 2: Class Diagram

## Activity Diagram

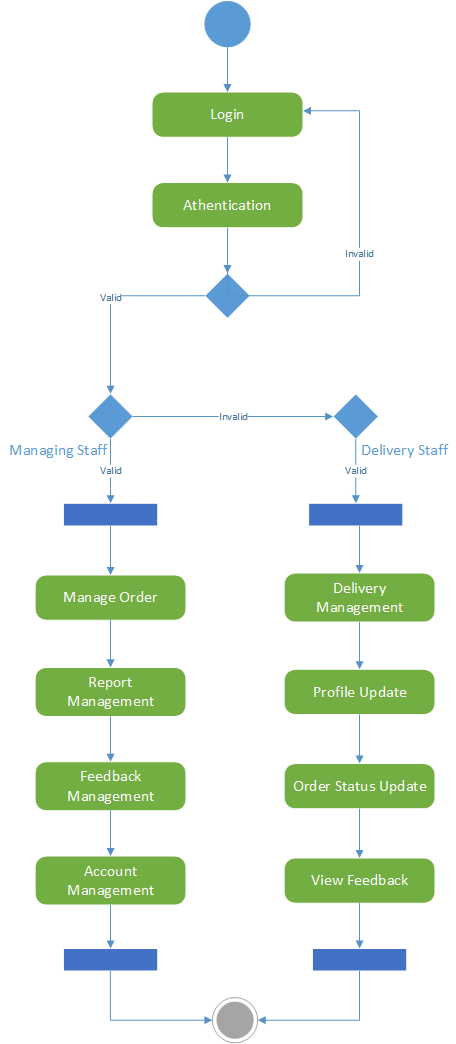
Activity Diagram is a diagram that represent a flow of control in a system. It shows the actions similar as flow chart or data flow diagram (DFD). In use case diagram it can also describe the steps. In Figure 3. A sequential steps of Courier Service System is shows where, it have both initial and final state. (smartdraw, 2021)

Figure 3: Activity Diagram

## GANTT chart

Gantt chart is the graphical or visual representation of the activities against the time where work is broken down in to several small which make easy to understand. ( Economic Times, 16 August, 2021)

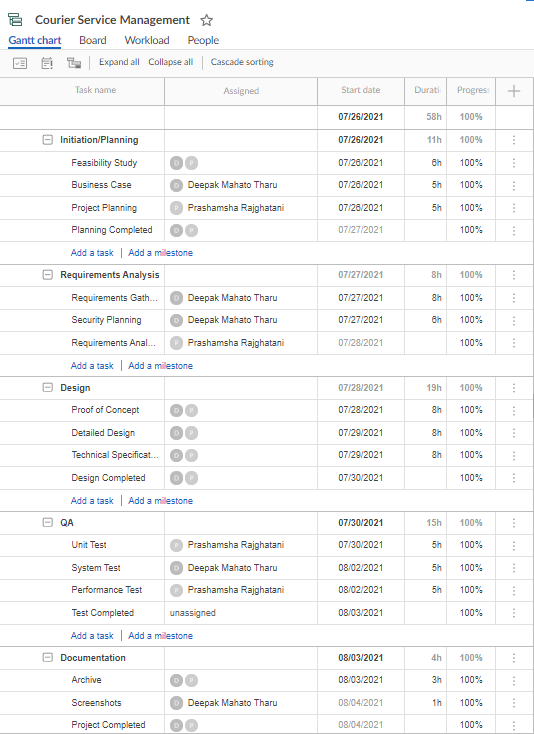


Figure 4: Gantt chart Timeline

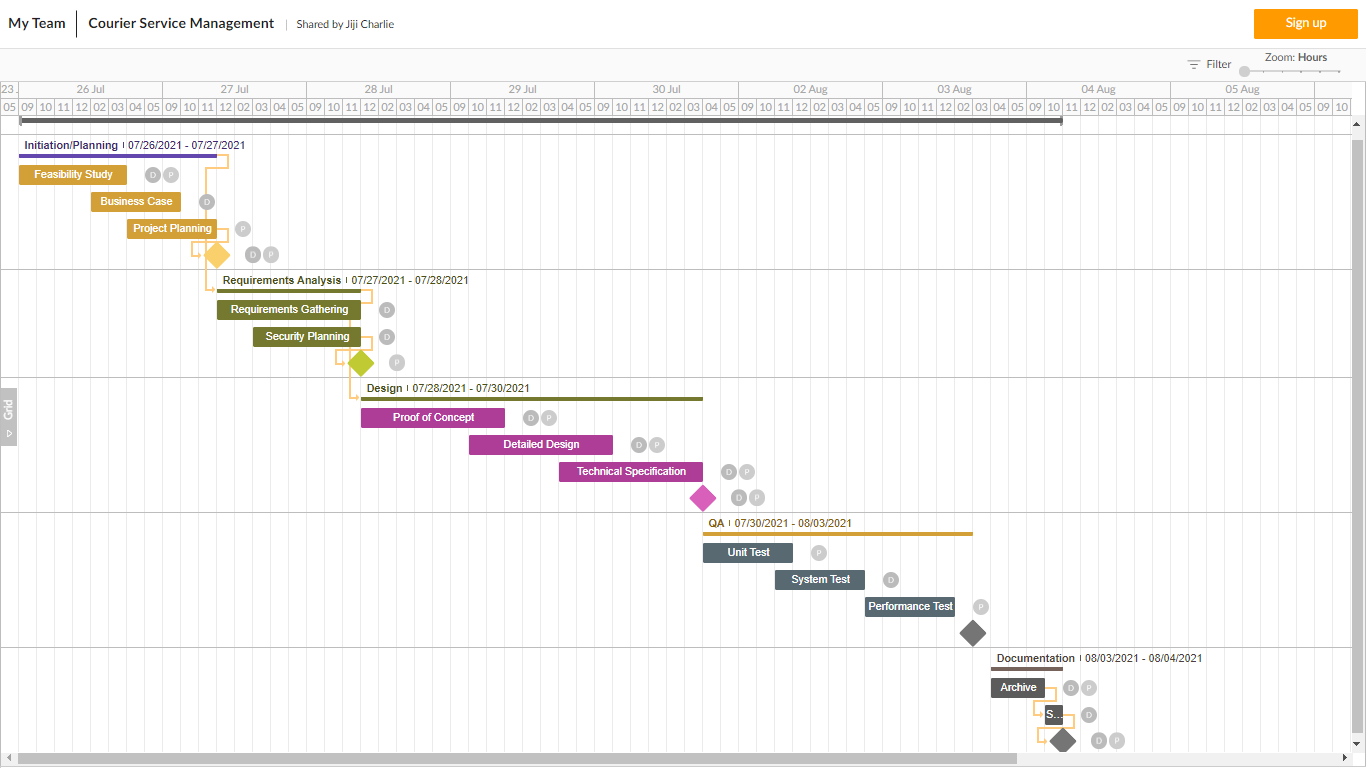
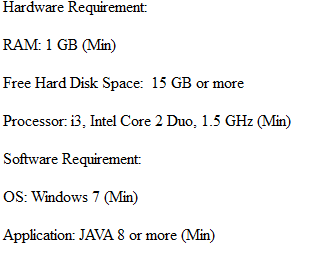


Figure 5: Gantt chart Timeline 2

# Hardware and Software Requirement



# Backup and Recovery

Backup and recovery refers to storing the data by creating copies of it so that we can extract it even after damage or loss of those. These are one of the crucial parts of software as if there occurs loss of any data then the user and owner can retrieve those data from backup readily. Data recovery is also equally important for the software.

In this developed online courier services management system as a source of backup and recovery we have made use of .txt files. A basic .txt files is used to store all the data and information that is generated from our software. Some of the files that we have used for backup and recovery are User Detail, Order Detail, Delivery Detail, and so on. (NetApp, 2021)

# Design and Implementation

## Class

A class refers to the blueprint or prototype that is user defined and used to create the object. Set of properties are denoted by class which are common to all types of objects of one type. Modifier, Class keyword, Class name, Superclass, Interface and Body are the components represented by the class declaration. For example, birds.



Figure 6: Class

## Object

An object is created from the class which represents real life entities. These are the real world things. State, Behavior and Identity are the components of the object. For example an online courier system might have objects like customers, order and menu.



Figure 7: Object

## Encapsulation

Encapsulation acquired when an object maintains a private state of all important data while only disclosing selective data and information to the outside world all inside an object. It adds a security layer to the system in an organized way. It can be private and public. (educative, 2020)

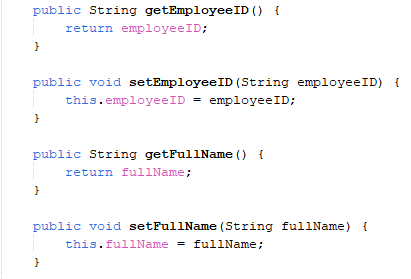


Figure 8: Encapsulation

## Constructor

Constructor in java refers to method that is used for initializing the newly created objects. It is called right after allocation of memory to an object. At the time of creating an object constructor is used to initialize the desire values. (Guru99, 2021)

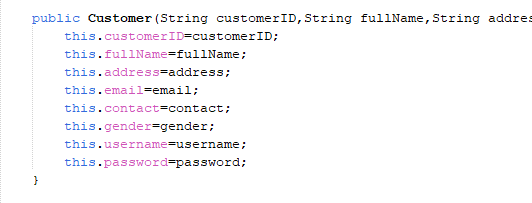


Figure 9: Constructor

## Inheritance

Inheritance is the ability to inherit the properties of other classes. This is popularly known as parent-child class where the child class inherits the properties of the parent class. Parent-child concept is brought to support reusability of the code, which helps to reduce code for overall programs. (javatpoint, 2021)

Figure 10: Inheritance

## Interface

In java an interface refers to the blue print of class with static constants and methods. It only can have method but not method body and is used to obtain the abstraction and multiple interface. (javatpoint, 2021)

Figure 11: Interface

Figure 12: Implementation

## Polymorphism

Polymorphism in object orientated programming is one of the feature that enable user performing a single task in different ways. (GeeksforGeeks, 2020)

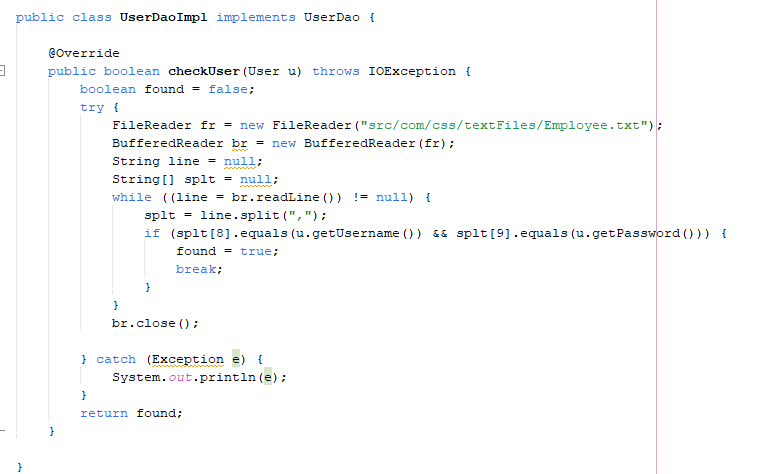


Figure 13: Polymorphism

# Screenshot

The system consist of mainly two user for login into system .i.e. managing staff and delivering. Interfaces design and developed for the system are listed below.

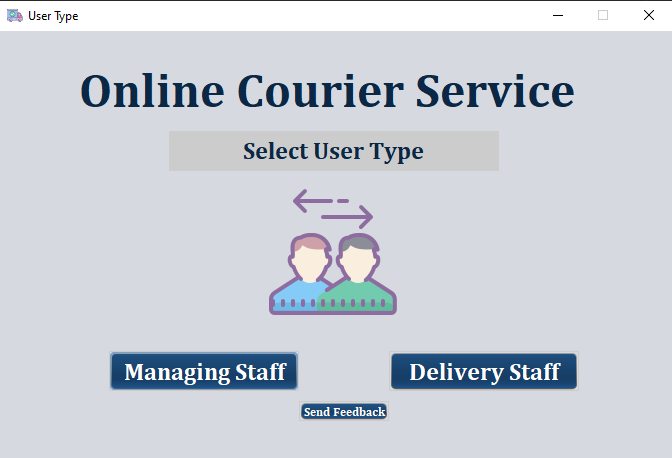


Figure 14: User Type

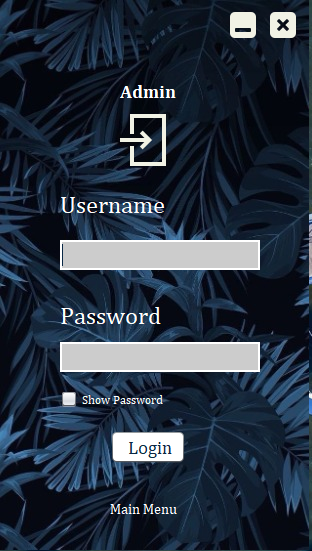
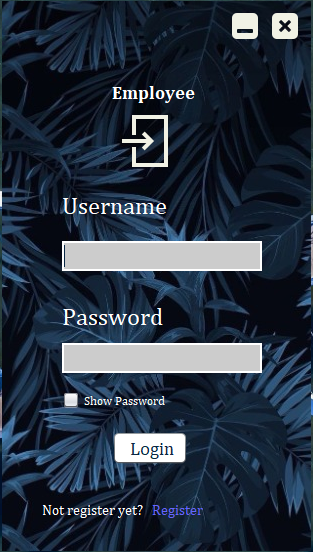


Figure 15: Login for Admin

Figure 16: Login for Employee

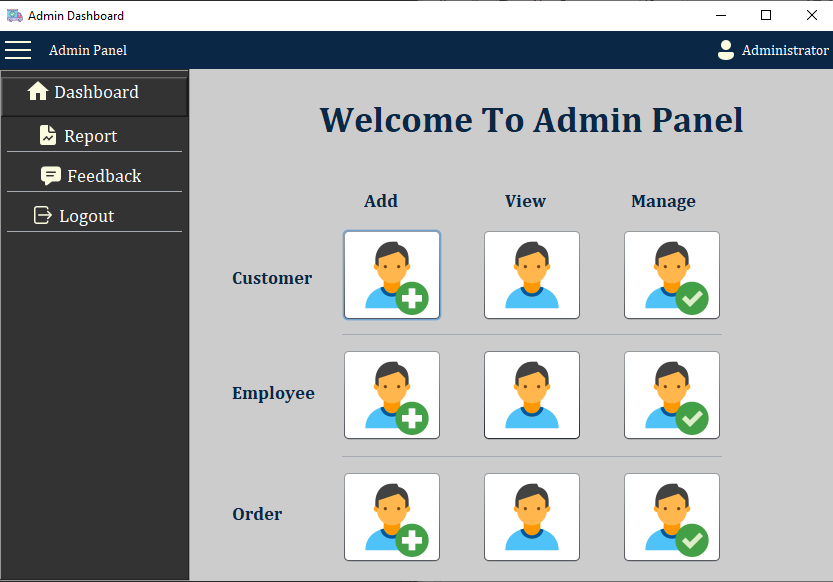


Figure 17: Admin Dashboard

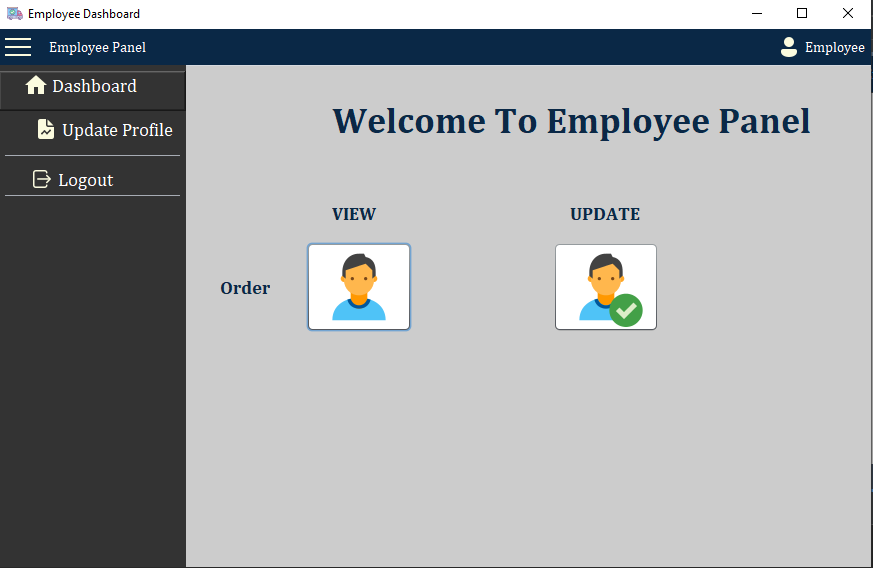


Figure 18: Employee Dashboard

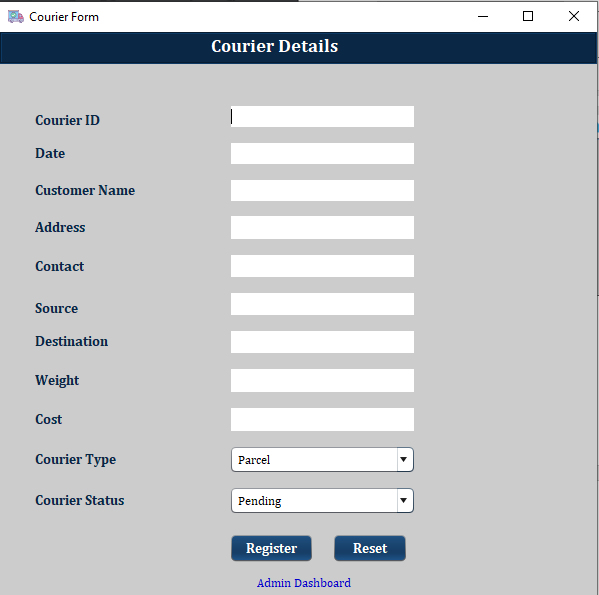


Figure 19: Courier Details Form

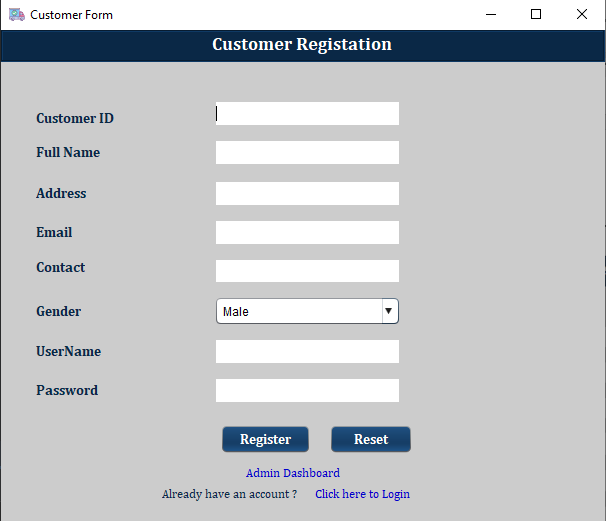


Figure 20: Customer Registration Form

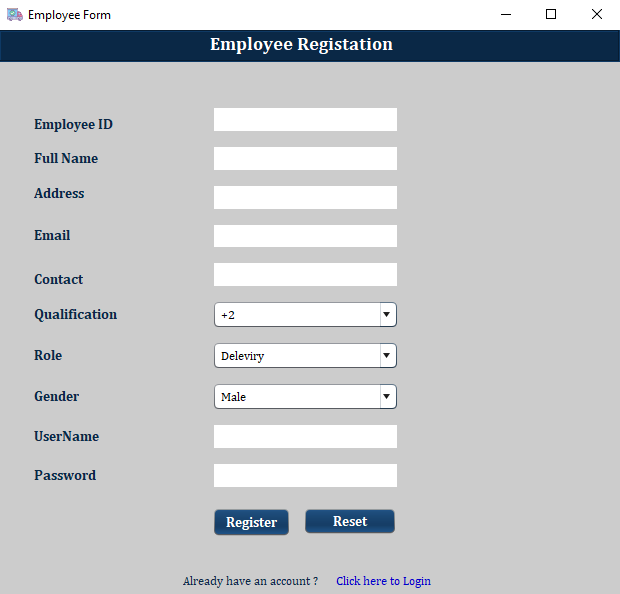


Figure 21: Employee Registration Form

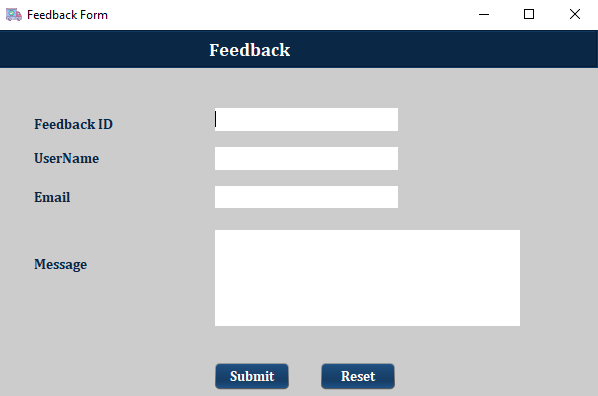


Figure 22: Feedback

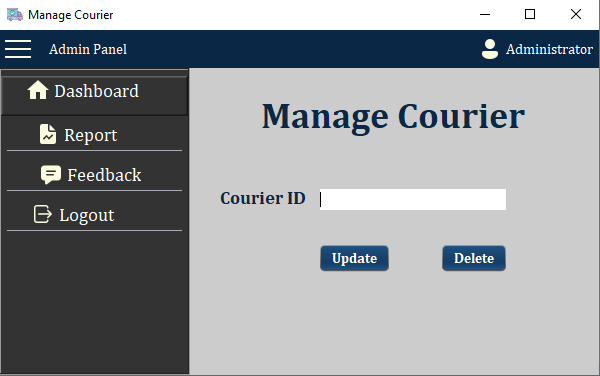


Figure 23: Manage Courier

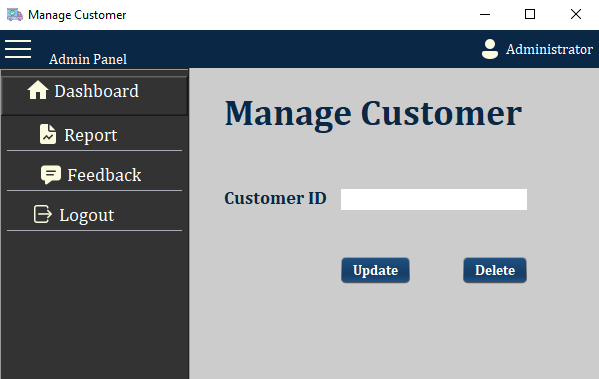


Figure 24: Manage Customer

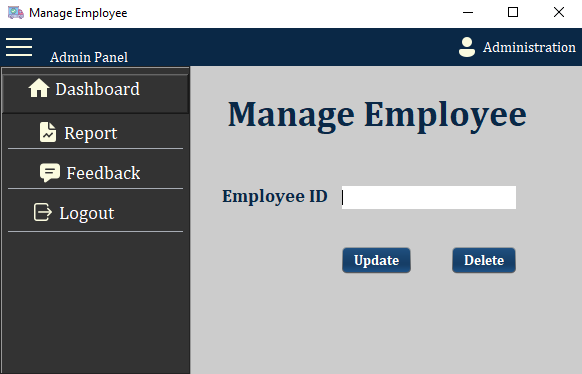


Figure 25: Manage Employee

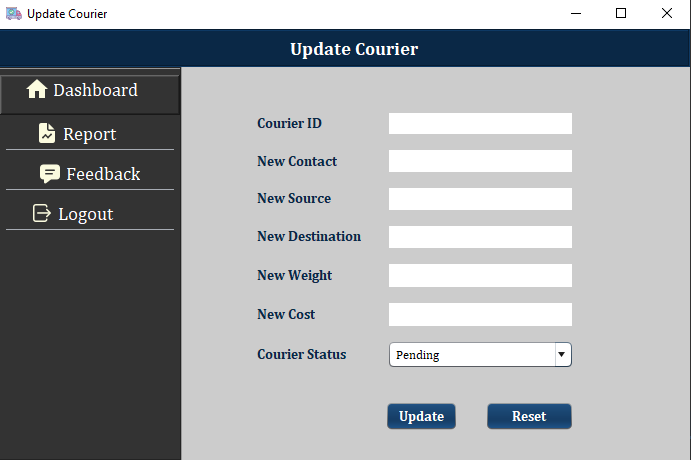


Figure 26: Update Courier

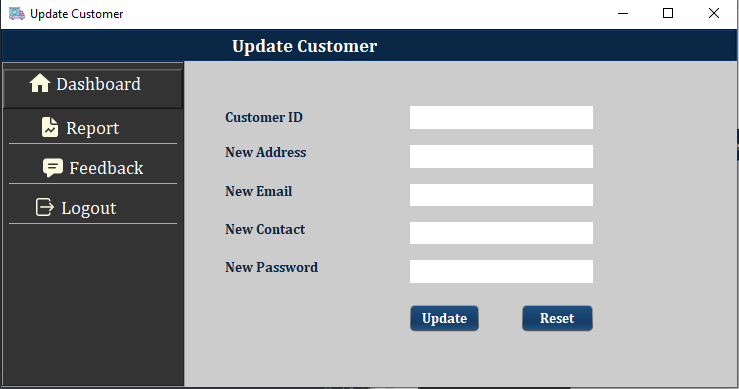


Figure 27: Update Customer

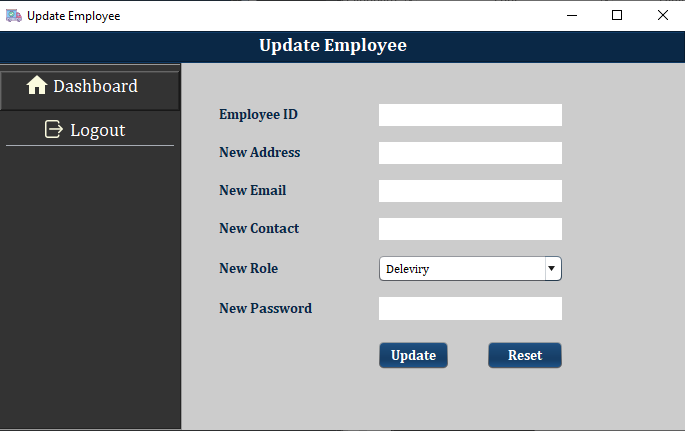


Figure 28: Update Employee

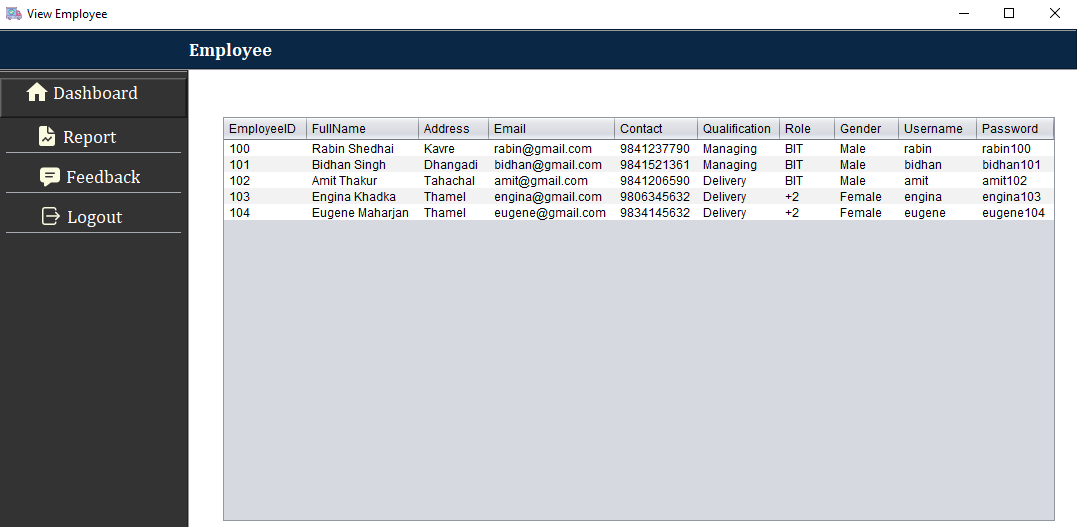


Figure 29: Employee Details

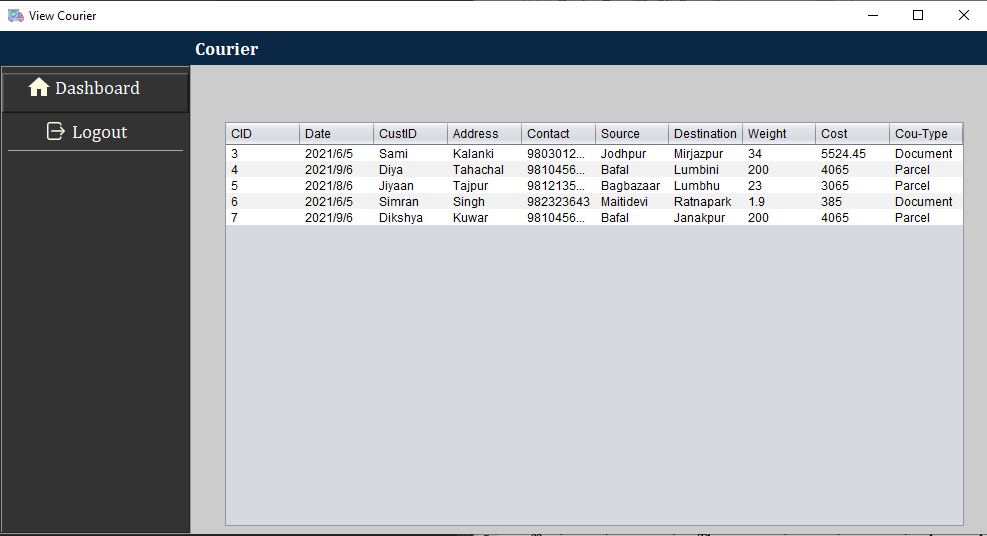


Figure 30: Courier Details

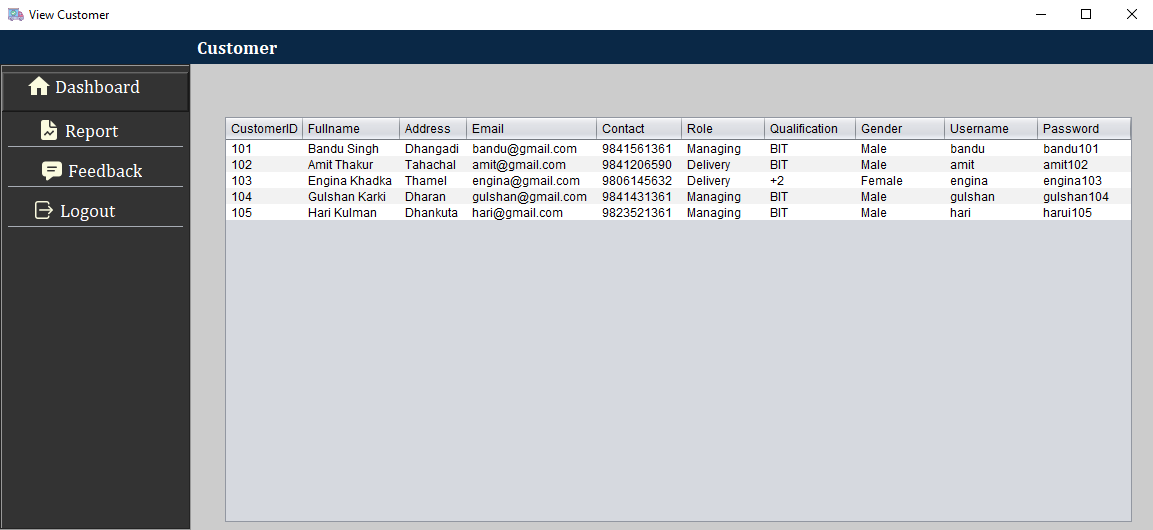


Figure 31: Customer Details

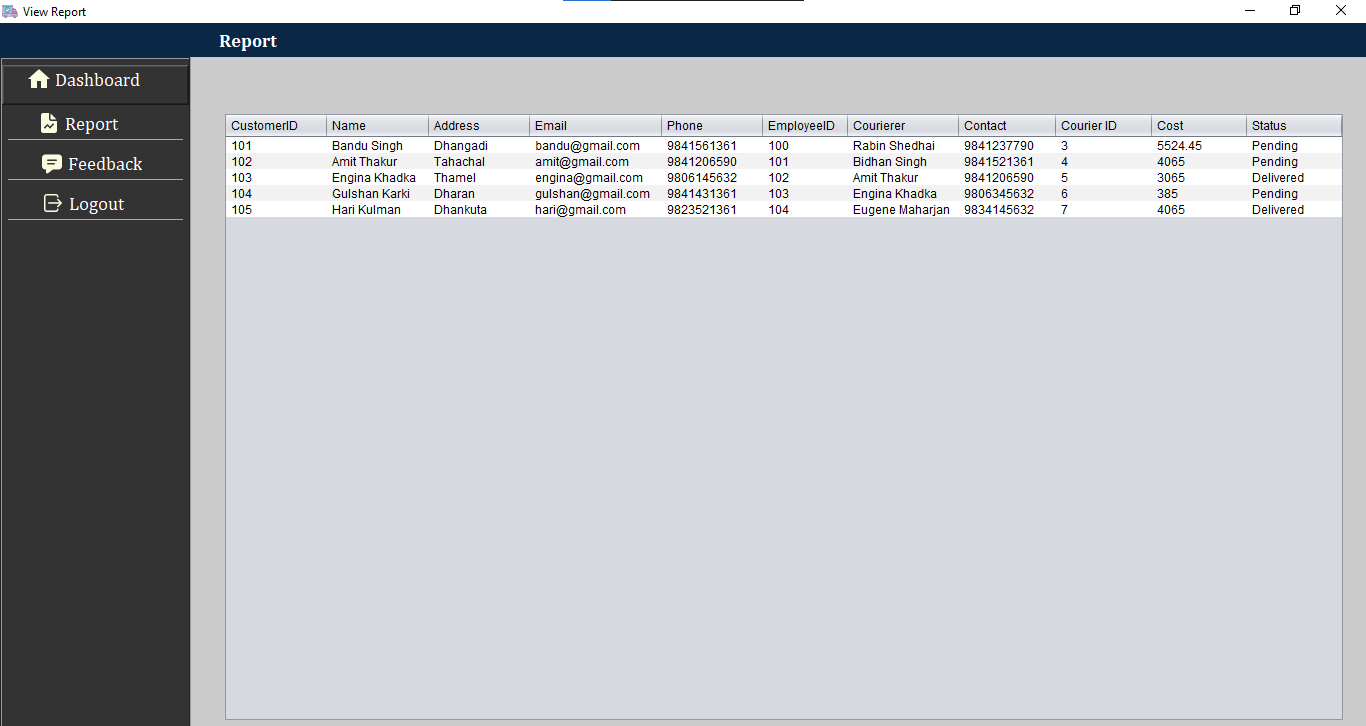


Figure 32: Report

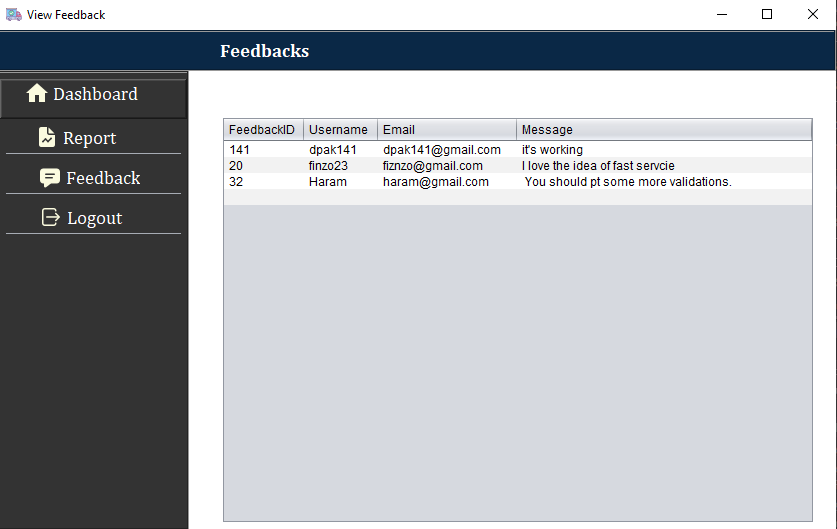


Figure 33: Feedback

# Testing Strategies

Testing is a very crucial part while developing software in order to ensure the proper functioning of the system. Testing is done by the developer so they should be able to choose effective testing strategies. There are various testing strategies that can be performed with newly developed software. Some of the testing strategies we performed with our project are listed are listed and described below:

## Unit Testing:

Uniting testing is initial and important testing performed by the developer to ensure every piece of the software is working properly. It makes debugging easier in the program as errors are discovered at a very early stage. This saves both time and cost to fix the errors. Unit testing is performed for our developed online courier service system where we review every module to check if every particular module is working correctly or not. (Software Development, Design and Coding, 2017)

## System Testing:

System testing is another testing strategy which is performed in order to evaluate a completed and integrated system. This testing ensures that it meets all the stated requirements of the software.  In this testing functionality of the program is tested from starting to end before delivering the software to the customer. It is conducted by a separate testing team. Through this testing we found that our online courier service system meets all the requirements including functional, commercial and technological. (Testing Code Security, 2007)

## Performance Testing:

Performance testing is testing that checks whether the developed system meets the performance requirements of the software or not. This type of testing is performed to see how a software behaves in different circumstances. The main goal of the testing is to see how responsive and stable software is in real user situations. Performance testing is done in every phase in the testing process. In our online courier service system we conducted this testing and concluded that performance of the system is very good. (Patterns for Performance and Operability: Building and Testing Enterprise Software, 2007)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. N | Test Description | Steps followed | Expected System Response | Status  (Pass/Fail) |
| 1 | Login | Login as Admin and Customer using RMI | All the user successfully login to system. | Pass |
| 2 | Register Customer  (**Create**) | Click on Sign Up to Register. Fill all the required details and click on register. | Receive successfully register. | Pass |
| 3 | View Food, Order and Customer (**Read**) | Login as Admin, customer and click on view food, order | All the details of admin, customer and food are listed | Pass |
| 4 | Update Order, Password  (**Update**) | Login as Admin, click on the update button as per the need respectively and change the required field and click on Update. | Required details were updated. | Pass |
| 5 | Delete Order from the order (**Delete**) | Logged in as required customer and deleted orders. | Order was deleted successfully. | Pass |
| 6 | Logout | Click on sign out | The user is logged out. | Pass |

# 

# Conclusions

Online Courier service system is developed as the java application for the easy and reliable courier. This system is design in such a way where customer and employee can be interacted and manage, everything under courier can be managed and customer can place couriers.

This Online system for the courier is design and developed using Apache NetBeans IDE where concepts of file handling is being used for the storage of the entry data. One of the challenging part while working for the development of this application was file handling part as all the storage of the data have to be done in text file it become quite difficult implement the read and write function in the system.

To sum up all the requirement for the system has been fulfill with some extra features like validation for the form, a responsive user interface which make system more effective and efficient to be used by users.

# References

**Books**

(2007). In M. A. Linden, *Testing Code Security* (p. 292). Auerbach Publications.

(2007). In C. Ford, *Patterns for Performance and Operability: Building and Testing Enterprise Software* (p. 340).

(2017). In J. F. Dooley, *Software Development, Design and Coding* (p. 330). Apress.

**Websites**

Economic Times. (16 August, 2021, August 16). *Definition of 'Gantt Chart'*. Retrieved from The Economic Times: https://economictimes.indiatimes.com/definition/gantt-chart

educative. (2020, Apr 15). *What is Object Oriented Programming? OOP Explained in Depth*. Retrieved from educative: https://www.educative.io/blog/object-oriented-programming

GeeksforGeek. (2020, Jun 29). *Types of Feasibility Study in Software Project Development*. Retrieved from GeeksforGeek: https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/

GeeksforGeeks. ( 2020, Dec 01 ). *Polymorphism in Java*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/polymorphism-in-java/

Guru99. (2021). *What is Constructor in Java?* Retrieved from Guru99: https://www.guru99.com/java-constructors.html

Java-Programming. (2021). *Class diagrams*. Retrieved from Java-Programming: https://java-programming.mooc.fi/part-11/1-class-diagrams

javatpoint. (2021). *Interface in Java*. Retrieved from javatpoint: https://www.javatpoint.com/interface-in-java

javatpoint. (2021). *Interface in Java*. Retrieved from javatpoint: https://www.javatpoint.com/interface-in-java

NetApp. (2021). *What Is Backup and Recovery?* Retrieved from NetApp: https://www.netapp.com/data-protection/backup-recovery/what-is-backup-recovery/

smartdraw. (2021). *What is an Activity Diagram?* Retrieved from smartdraw: https://www.smartdraw.com/activity-diagram/

tutorialspoint. (2021). *UML - Use Case Diagrams*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/uml/uml\_use\_case\_diagram.htm