# COURIER MANAGEMENT SYSTEM

*FINAL REPORT*

*Degree of*

### BACHELOR OFTECHNOLOGY

**in**

#### COMPUTER SCIENCEAND ENGINEERING

By

|  |  |  |
| --- | --- | --- |
| **Registration No.** | **Name** | **Roll No.** |
|  |  |  |
|  |  |  |
|  |  |  |

Under the guidance of

**AMARINDER KAUR**



#### School of Computer Science and Engineering

Lovely Professional University Phagwara, Punjab (India) APRIL 2023

Page | 1

### ACKNOWLEDGMENT

Place: Lovely Professional University Date: 10th APRIL, 2023

We would like to thank Ms. AMARINDER KAUR for assigning us with this project. Through the project we can grasp more technical and have a hands-on practical experience with Java projects. Through it we can learn how a project is created and how necessary and crucial technical knowledge is. We are grateful to the faculty that has provided us with the necessary guidelines.

|  |  |
| --- | --- |
| **Name** | **Registration No.** |
|  |  |
|  |  |
|  |  |

Page | 2

### DECLARATION STATEMENT

Place: Lovely Professional University Date: 10th APRIL 2023

This is to declare that this report has been written by us. No part of the report is copied from the other sources. All information included from other sources has been duly acknowledged. We are aware that if any part of the report is found to be copied, we will take full responsibility for it.

Page | 3

**Introduction**

The "Courier Management System" is a Java console application that aims to provide a simple and interactive management system for tracking courier shipments. The system allows users to create a shipment, track a shipment, view all shipments, and view the invoice. The system also allows administrators to view all shipments.

### Overall structure

The program begins with importing some necessary classes such as Scanner and Random. Then, it declares some static variables such as id, name, destination, locations, newLocations, and packageWeight. It also defines an array for storing admin names and their passwords. Finally, it defines the main method, which contains a menu-driven user interface.

### Overview of Code

The code starts with importing two packages, java.util and java.util.Random, and then declares several variables such as Scanner, Random, Integer, String, and Double. It also initializes some variables, such as id and Locations with random numbers and location names, respectively.

The code uses a do-while loop to display the main menu options to the user until the user chooses to exit the application. The switch statement is used to handle each menu option, and the appropriate method is called accordingly.

Page | 4

### FUNCTIONS

The program has a main method that displays a menu with the following options:

1. Create Shipment

2. Track Shipment

3. View All Shipments (For Admin)

4. View Invoice

5. Exit

6. Return to Main Menu

**Create Shipment**

The addPackage() method allows the user to create a shipment by providing the name, weight, and destination of the package. The method first checks if the weight is valid and within the range of 0 to 50 KG. If the weight is invalid, the user is prompted to enter a valid weight. If the weight is above 50 KG, the user is notified that the package cannot be delivered.

The destination entered by the user is checked against a list of available locations. If the location is not available, the user is asked to pay extra charges to send the package to the destination. If the location is available, the user is assigned a unique tracking ID, and the package is added to the system.

Page | 5

# 

**Track Shipment**

The trackPackage() method allows the user to track a shipment by providing the tracking ID. The method generates a random time for the delivery and notifies the user of the expected delivery time. If the tracking ID is not found, the user is notified that the package was not found.

# View All Shipment (For Admin)

# The view\_All\_Shipments() method allows the administrator users to view all shipments. The administrator is prompted to enter their ID, and if the ID is correct it display the message for enter the Valid ID and if password is correct , the method displays all shipments in the system with their Tracking Id, Address,Cost and further Details.

**View Invoice**

The viewInvoice() method allows the user to view the invoice for a specific package by entering its tracking ID. This Function Displays all Information related to the the specific package.

**Exit**

It allows the user to exit the program.

## **Coding Section**

#### Add Pacakge()

This option allows the user to create a shipment. They are asked to enter their name, the receiver's name, the weight of the package, and the destination of the package. If the weight is invalid (less than or equal to zero, or greater than 50), the program prompts the user to enter a valid weight. If the destination is valid (i.e., one of the cities in the **Locations** array), the program generates a random tracking ID, adds the package details to the **packages** array, and adds the tracking ID to the **pkgid** array.

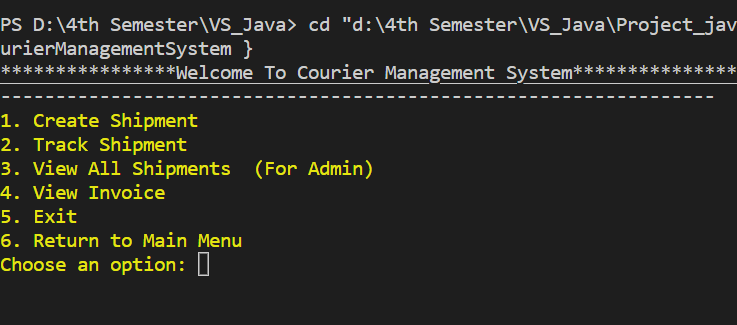
**Track Shipment()**  
  
  
  
This option allows the user to track a shipment. They are asked to enter the tracking ID of the package, and the program searches for that ID in the pkgid array. If the ID is found, the program prints the package details (sender’s name, receiver’s name, weight, destination, and tracking ID).  
It also print the remaining time for the package to arrive for the Destination. It makes this as it generates the random number between 1 – 10 as hour remaining for the package.  
   
  
  
  
 **View All Shipments (for Admin)**  
  
  
This option allows the admin to view all shipments. The admin is asked to enter their username and password. If the username and password are correct (stored in the **Admin** and **Admin\_Pass** arrays), the program prints the details of all packages (sender’s name, receiver’s name, weight, destination, and tracking ID).  
  
For looking our Project more attractive we have added some animations using thread Handling

Making the delay time for each character so it looks like LOADING animation.  
  
  
 **Remove\_all\_Shipments()**

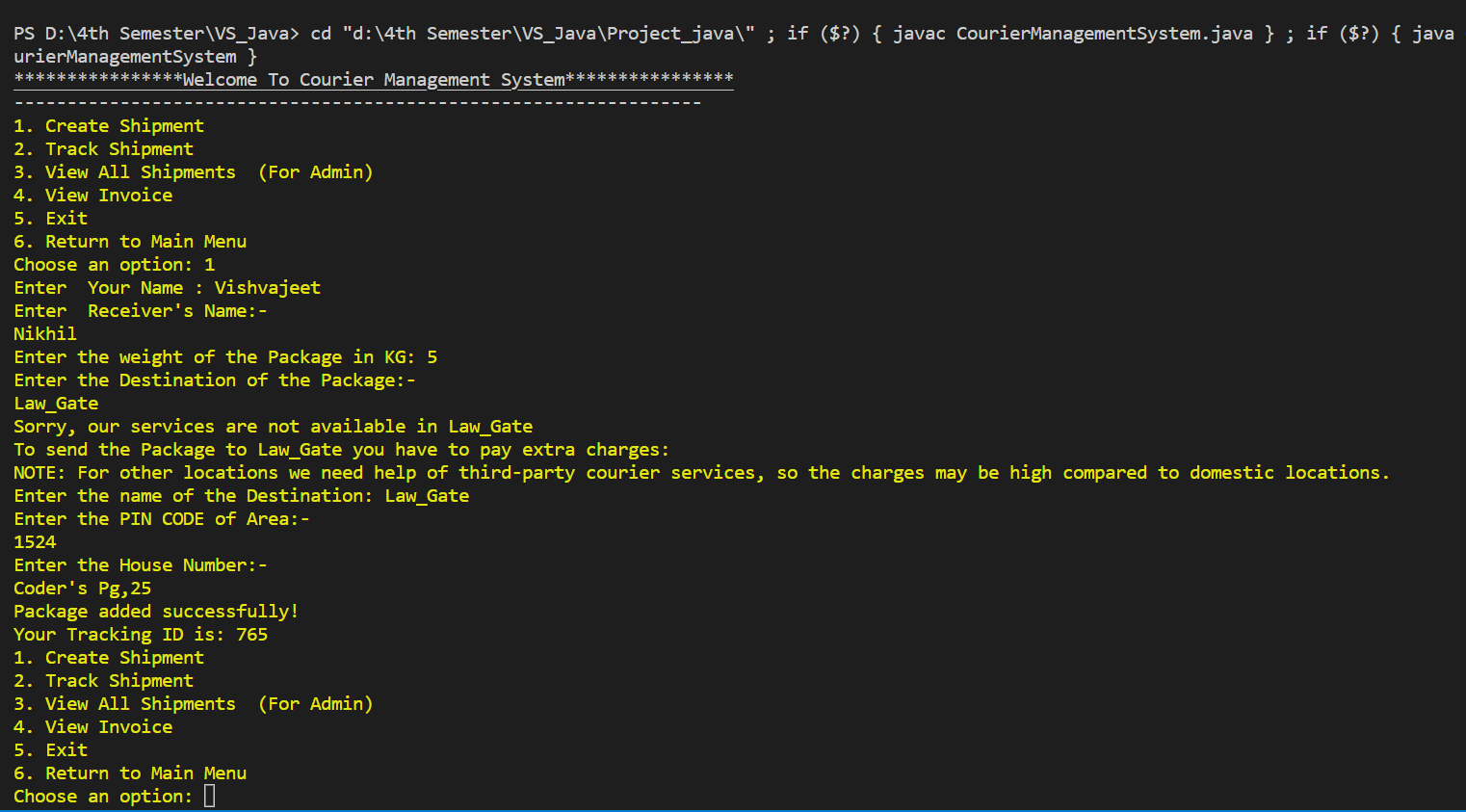
This option is available for admins only when admin login to their account they got 2 options first one is View all shipments and second one is removing all Shipments. This function deletes all the shipments that are previously created by the user.

### View Invoice()

This option allows the user to view the invoice for a shipment. They are asked to enter the tracking ID of the package, and the program searches for that ID in the **pkgid** array. If the ID is found, the program calculates the cost of shipping based on the weight of the package and the destination (using a fixed rate per kilogram and an extra charge for destinations not in the **Locations** array) and prints the invoice (sender's name, receiver's name, weight, destination, tracking ID, and cost).  
  
  
  
 Exit()  
  
  
This option allows the user to exit the program with exit() method

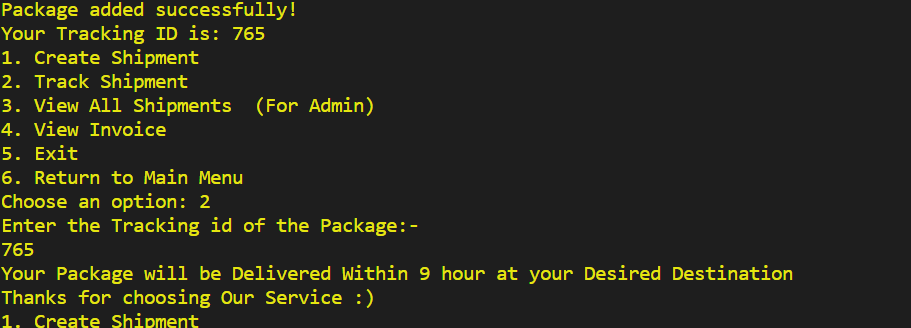
  
Here are some of the Screen-captures that helps us to understand the program in more detail  
  
  
  
  
 Home Page of the Program

User must enter his choice by Pressing 1,2,3,4,5 & 6 according to his/her Requirement.

Every number represents the program that is written in front of the number.   
  
  
  
  
  


Demo of creating the pacakge

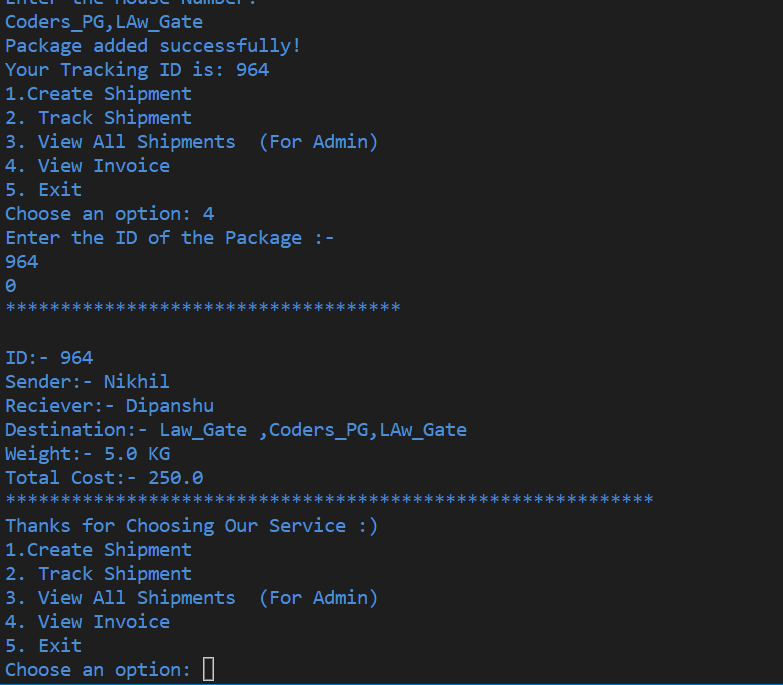
Unique package ID for tracking the package



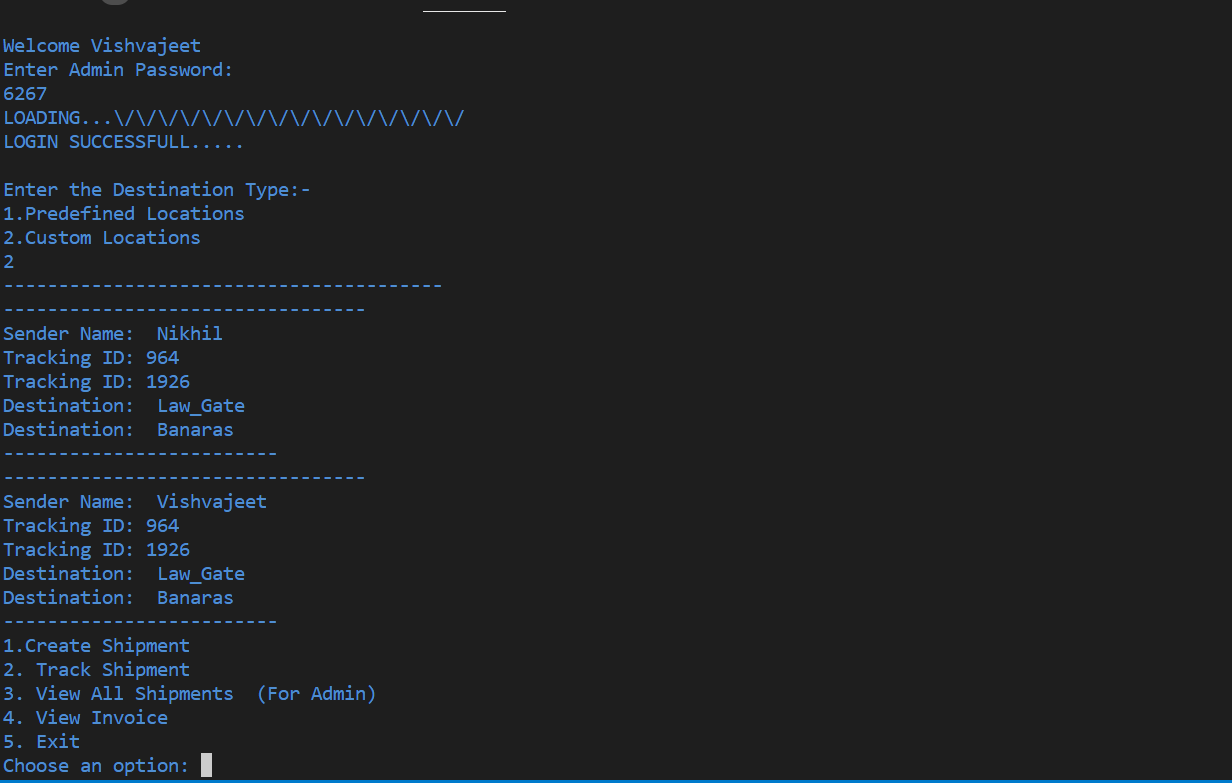
here’s how user will Track his Package

**Return to Main Menu**

It allows the user to return to the main menu.

Page | 7

Invoice of the package



Admin Section

Information about all Packages

**Teamwork**   
  
  
  
  
   
 Our group consists of three members who have been assigned different roles based on their

skills and interests. Vishvajeet is responsible for the coding and algorithm part of our project. He

has a strong background in programming and has extensive experience in designing and

implementing algorithms. He will be instrumental in ensuring that our project is technically sound

and performs optimally.

Nikhil is responsible for suggesting functionalities and managing the project. He has a keen eye

for detail and a knack for identifying areas where the project can be improved. He will also be

responsible for keeping the project on track and ensuring that all members are meeting their

deadlines.

Dipanshu is responsible for the report work and managing the Project on Github. He has strong

communication skills and can present information in a clear and concise manner. He will be

responsible for creating reports, documenting the project's progress, and ensuring that all members are

aware of any updates or changes.

Overall, our group will work collaboratively to ensure that our project is successful. We will communicate

regularly to ensure that everyone is aware of their roles and responsibilities, and we will work together to

overcome any challenges that may arise. Each member will be accountable for their work and will be

expected to meet their deadlines to ensure that the project is completed on time.

**References**

1. GeeksforGeeks
2. StackOverflow
3. JavaForums
4. codewithHarry

| 10