Project Report

Project Title:

"Courier Service Management System"

Course Code: CSE246

Course Title: Algorithms

Semester: Spring24

Section: 07

Submitted By:

NO.	Name	ID
01.	Abdul Mumeet Pathan	2022-1-60-267
02.	Tasnuva Tasnim Nova	2022-1-60-266
03.	Taniz Fatema Jarin	2022-1-60-065

Submitted To:

Amit Mandal

Lecturer

Department of Computer Science and Engineering

Submission Date: **26 May, 2024**

1. Introduction

This project optimizes delivery routes, ensuring timely and cost-effective package handling. The system will leverage three key algorithms:

- **Rabin-Karp Algorithm:** The Rabin-Karp algorithm in our code is a string-searching algorithm that uses hashing to find a pattern within a text. In this code, the Rabin-Karp algorithm is used to compare the hashed version of the entered password with the hashed version stored in the user or delivery person account file accordingly. If the hash values match, it further checks the actual characters to confirm the match.

- **Dijkstra's Algorithm:** This algorithm finds the shortest path between a source (origin) and all destinations (customer addresses), ensuring the most efficient delivery routes for couriers.

- **0-1 Knapsack Problem:** This algorithm optimizes the selection of packages for a courier to carry, considering weight constraints. It ensures couriers can maximize deliveries while adhering to capacity limitations.

2. Problem Statement

The challenge is to design an algorithm-based system for a courier delivery system. This system optimizes delivery routes to minimize travel time and distance for couriers, while ensuring efficient package allocation considering capacity constraints. The challenge lies in designing a system that addresses the following key aspects of courier delivery:

- **Route Optimization:** Identifying the most efficient delivery routes that minimize travel time and distance for couriers.

- **Cost Efficiency:** Ensuring cost-effectiveness by optimizing delivery routes and courier capacity utilization.

3. Objectives

This project aims to achieve the following objectives:

- Develop an algorithm-based system that optimizes delivery routes, minimizing travel time and distance.

- Allow users to place orders and specify details like item weight and cost.

- Integrate the Rabin-Karp algorithm for efficient about information within the system.

- Implement Dijkstra's algorithm to find the shortest paths between the origin and destination of courier for the delivery.

- Employ the 0-1 Knapsack problem to optimize package allocation for couriers, considering capacity constraints.

- Enable delivery persons to view, accept, and manage orders based on their vehicle's maximum weight capacity.

- Ensure delivery persons can maximize their profit by selecting the optimal combination of items to deliver.

4. Methodology

The methodology of the system is outlined as follows:

1. User and Delivery Person Account Management:

- **Sign Up:** Users and delivery persons create accounts with unique usernames and passwords.
- **Login:** Authentication using the Rabin-Karp algorithm to ensure secure access.

2. Order Placement:

- Users can place orders by specifying the number of items, their weights, and profits.

3. Order Viewing and Acceptance:

- Delivery persons can view available orders and use the knapsack algorithm to determine if they can carry the order based on their vehicle's maximum weight capacity.

- Delivery persons accept orders that fit within their capacity.

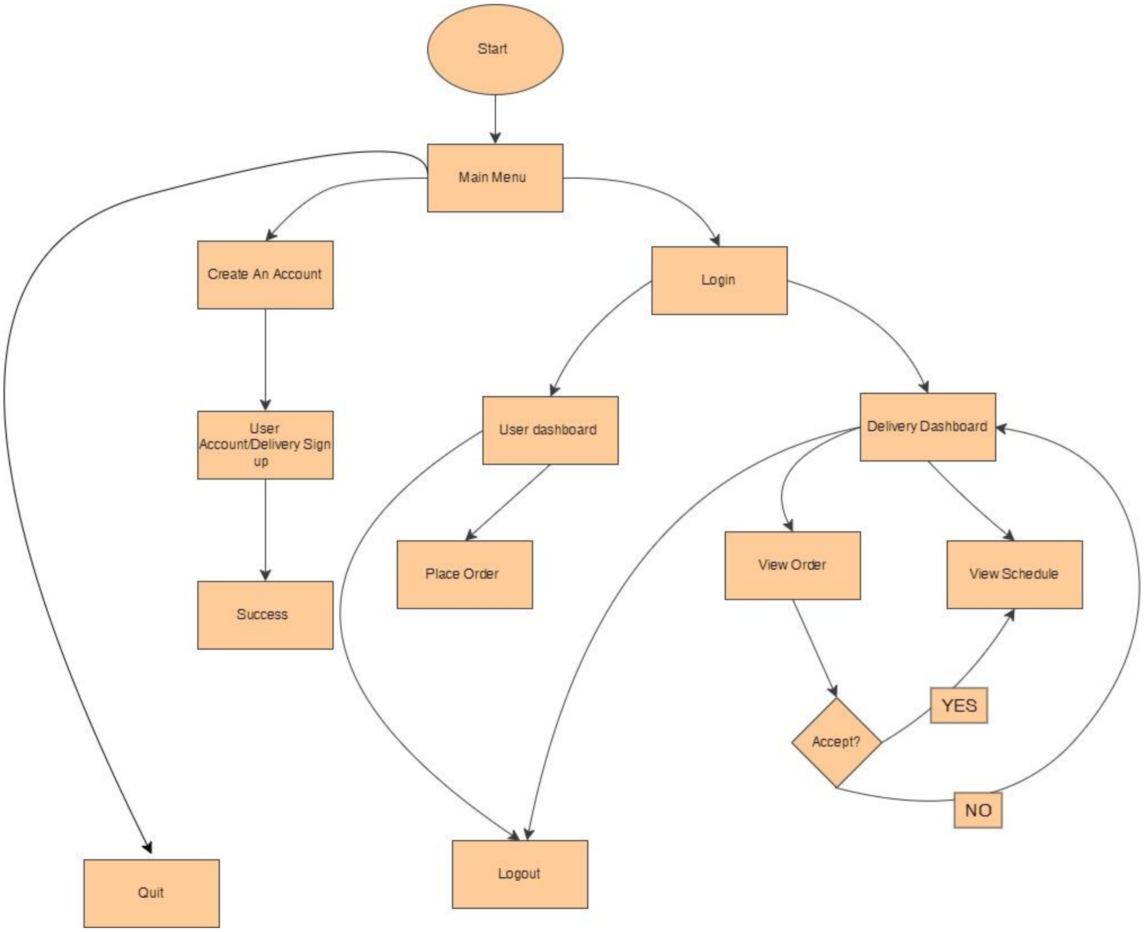
4. Route Optimization:

- Use Dijkstra's algorithm to find the shortest path for delivering orders, ensuring efficiency.

5. Order Management:

- Accepted orders are added to the delivery person's schedule.

Flowchart



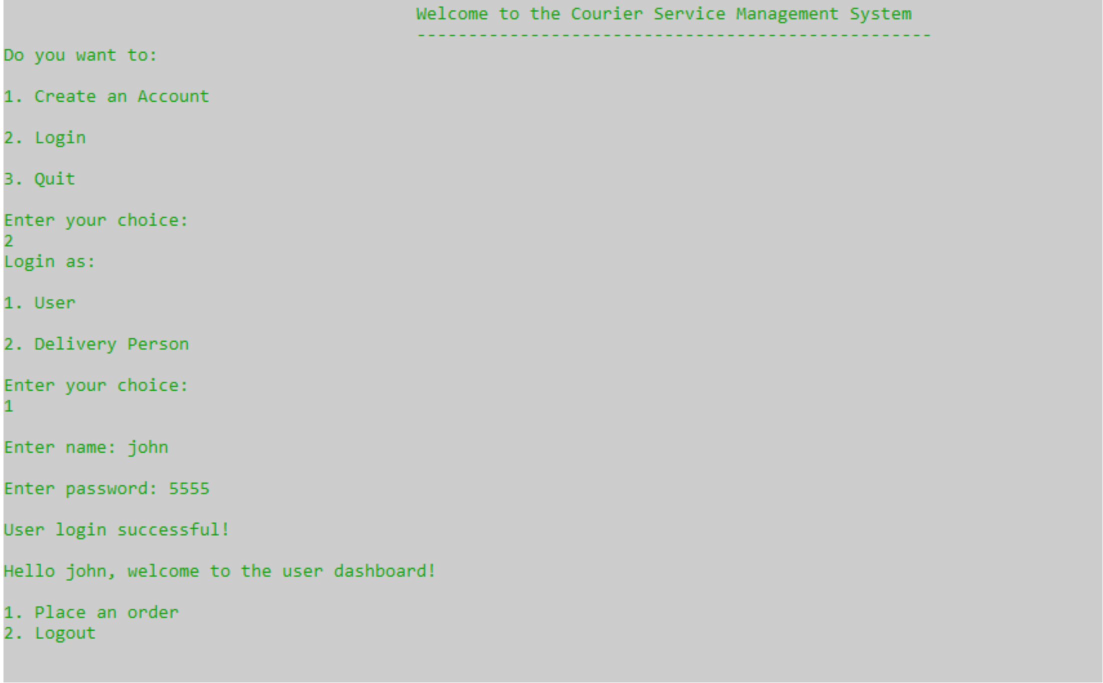
5. Limitations

- The adjacency matrix for Dijkstra’s algorithm is hard-coded.
- The system assumes one delivery route for simplicity, which may not be efficient in real-world scenarios with multiple orders.
- The 0-1 Knapsack algorithm assumes fixed capacity, which might not be flexible enough for varying package sizes and weights.
- The authentication mechanism relies solely on comparing hashed passwords using the Rabin-Karp algorithm. So for larger values it might be as issue.
- The command-line interface provided by the code is basic and may not offer a user-friendly experience.
- The code lacks comprehensive input validation and robust error handling mechanisms. Means it doesn't handle cases where file I/O operations failure can occur. Input validation needs to be performed to ensure that inputs are within expected ranges and formats.
- The code uses Windows-specific functions like SetConsoleTextAttribute and system("cls"), making it less portable across different operating systems.

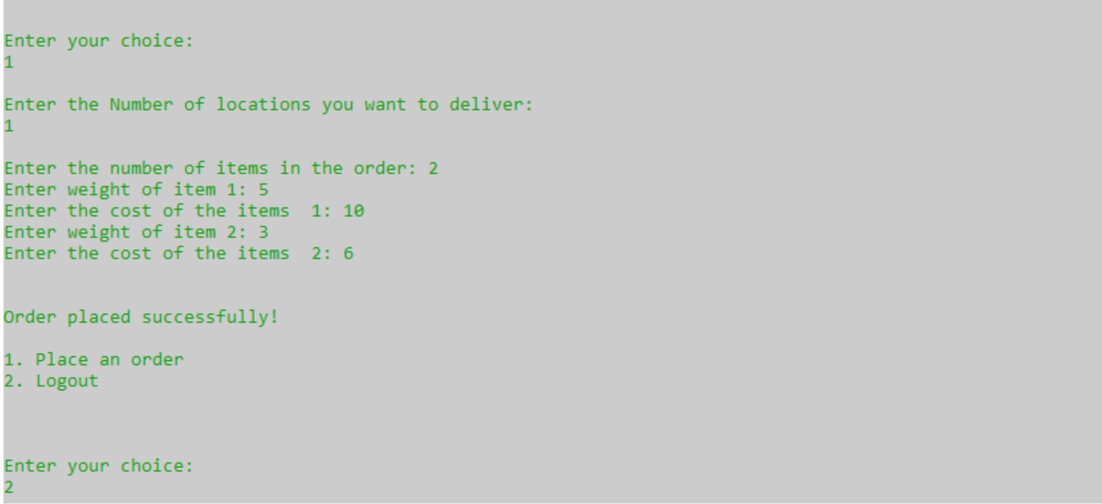
6. Result

The system successfully allows users to sign up, log in, and place orders. Delivery persons can view and accept orders based on their capacity, and the system calculates the optimal route for deliveries using Dijkstra’s algorithm. The knapsack algorithm ensures that delivery persons maximize their profit within the constraints of their vehicle’s capacity.

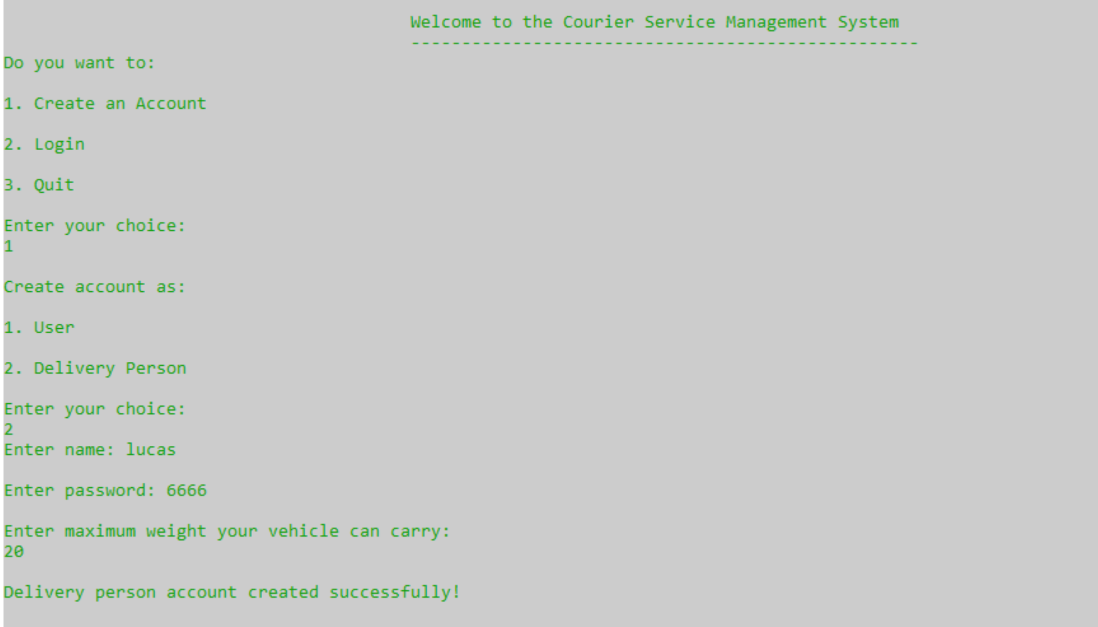
Outputs



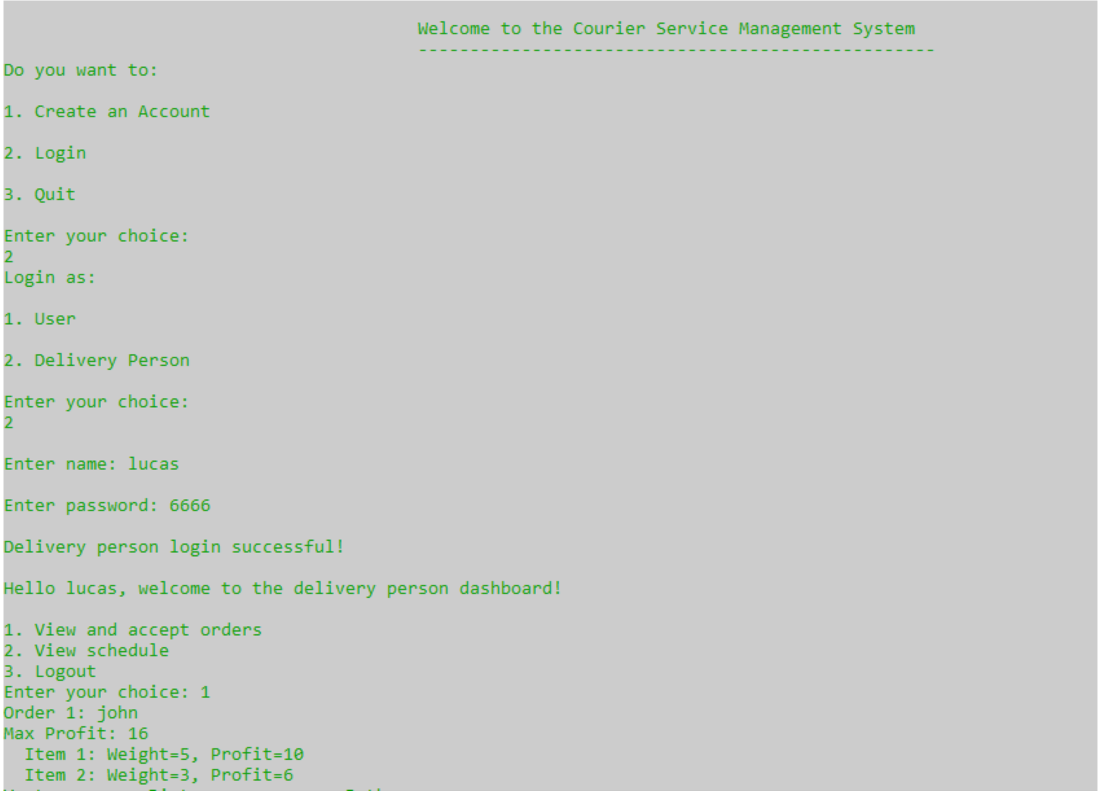
```
                                        Welcome to the Courier Service Management System
                                        --------------------------------------------------
Do you want to:

1. Create an Account

2. Login

3. Quit

Enter your choice:
2
Login as:

1. User

2. Delivery Person

Enter your choice:
1

Enter name: john

Enter password: 5555

User login successful!

Hello john, welcome to the user dashboard!

1. Place an order
2. Logout
```



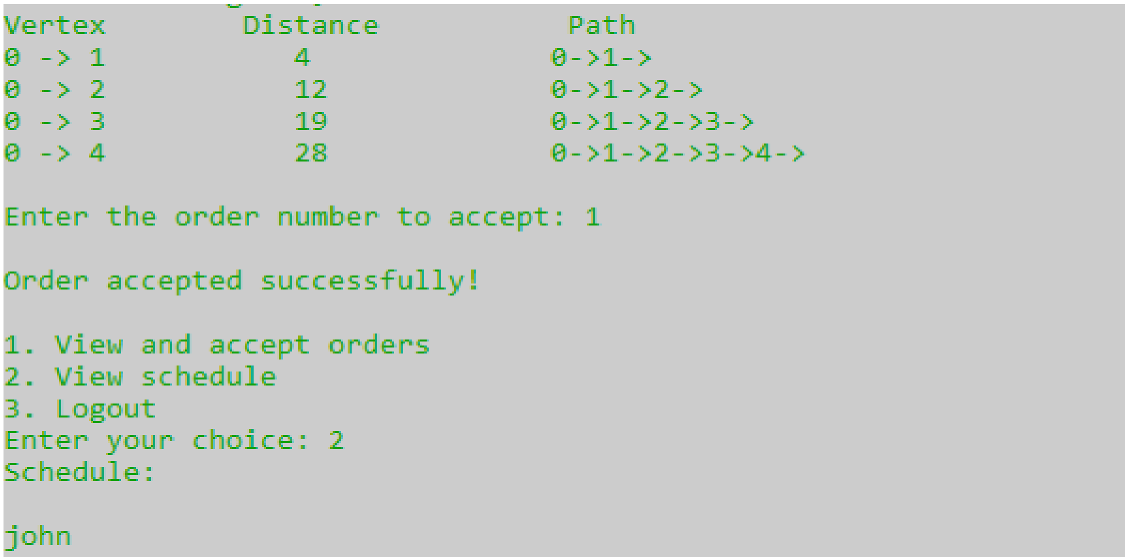
```

Enter your choice:
1

Enter the Number of locations you want to deliver:
1

Enter the number of items in the order: 2
Enter weight of item 1: 5
Enter the cost of the items  1: 10
Enter weight of item 2: 3
Enter the cost of the items  2: 6


Order placed successfully!

1. Place an order
2. Logout



Enter your choice:
2
```



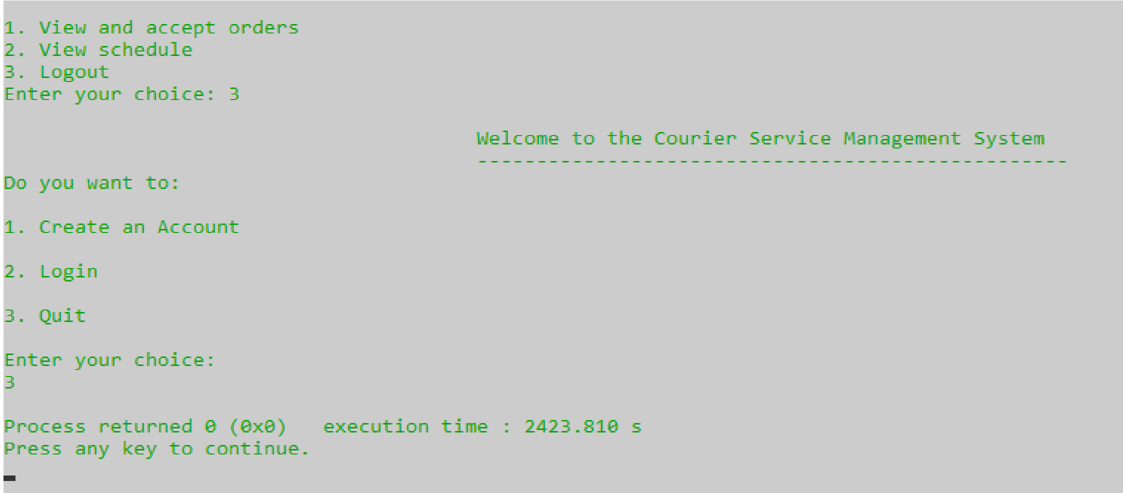
```
                                        Welcome to the Courier Service Management System
                                        --------------------------------------------------
Do you want to:

1. Create an Account

2. Login

3. Quit

Enter your choice:
1

Create account as:

1. User

2. Delivery Person

Enter your choice:
2
Enter name: lucas

Enter password: 6666

Enter maximum weight your vehicle can carry:
20

Delivery person account created successfully!
```



```
                                        Welcome to the Courier Service Management System
                                        --------------------------------------------------
Do you want to:

1. Create an Account

2. Login

3. Quit

Enter your choice:
2
Login as:

1. User

2. Delivery Person

Enter your choice:
2

Enter name: lucas

Enter password: 6666

Delivery person login successful!

Hello lucas, welcome to the delivery person dashboard!

1. View and accept orders
2. View schedule
3. Logout
Enter your choice: 1
Order 1: john
Max Profit: 16
  Item 1: Weight=5, Profit=10
  Item 2: Weight=3, Profit=6
```



```
Vertex         Distance            Path
0 -> 1            4              0->1->
0 -> 2            12             0->1->2->
0 -> 3            19             0->1->2->3->
0 -> 4            28             0->1->2->3->4->

Enter the order number to accept: 1

Order accepted successfully!

1. View and accept orders
2. View schedule
3. Logout
Enter your choice: 2
Schedule:

john
```



```
1. View and accept orders
2. View schedule
3. Logout
Enter your choice: 3

                                        Welcome to the Courier Service Management System
                                        --------------------------------------------------
Do you want to:

1. Create an Account

2. Login

3. Quit

Enter your choice:
3

Process returned 0 (0x0)   execution time : 2423.810 s
Press any key to continue.
```

7. Conclusions

The Courier Service Management System provides a functional prototype for managing courier deliveries. It combines user authentication, order management, profit maximization, and route optimization into a cohesive system. Future improvements focusing on enhancing security, scalability, and flexibility to handle a more extensive and dynamic set of real-world delivery scenarios can make the system evolve.

END