

# UNIVERSITY OF ENGINEERING AND TECHNOLOGY LAHORE (NEW CAMPUS)



## Courier Logistics Engine

### **Submitted By:**

Zeeshan Mir(2024-SE-13)

Usman Ahmad (2024-SE-22)

Ahmar Saleem (2024-SE-27)

Muhammad Fahad (2024-SE-31)

### **Submitted To:**

**Mr. Ali Raza**

### **Subject:**

Data Structures and Algorithms (Lab)

# Table of Contents

## Contents

Introduction .....	3
Problem Statement .....	3
Modules Explanation .....	4
Diagrams .....	5
Class diagram .....	5
System Architecture .....	6
Flowcharts .....	7
1. System Entry & Control Flow .....	7
2. Parcel Ingestion & Priority Sorting.....	8
3. Path finding & Dispatch Logic .....	9
4. Real-Time Status & Lifecycle .....	10
5. Operations, Undo, & Persistence.....	11
Screenshots:.....	12
Algorithm Choice & Time Complexity .....	14
Drawbacks .....	14
Links .....	15
GitHub Link.....	15
Linkedin Link.....	15
Conclusion .....	15

## Introduction

The Swift-Ex Intelligent Logistics System is a robust software solution designed to automate and optimize parcel delivery operations. In the modern logistics landscape, timely and accurate parcel tracking is crucial for customer satisfaction, operational efficiency, and effective resource management. This system integrates multiple functionalities such as parcel registration, route calculation, rider assignment, real-time transit monitoring, and historical tracking of each parcel. By leveraging efficient data structures and intelligent algorithms, the system ensures that parcels reach their destinations quickly and reliably while adapting to real-world challenges like road blockages, traffic congestion, and overloaded delivery routes.

A key feature of the Swift-Ex system is its modularity and scalability. Each core functionality—parcel management, rider allocation, route optimization, and live tracking—is encapsulated within dedicated classes and data structures. This design allows for easy maintenance, testing, and future expansion, such as adding new delivery rules, cities, or parcel types. Users interact with the system through a menu-driven interface, which provides clear and organized access to all functionalities, allowing logistics managers to process new deliveries, monitor ongoing shipments, and analyze historical data efficiently.

The system also emphasizes real-time responsiveness. Parcel statuses are continuously updated, and route calculations are dynamically adjusted based on changing conditions. This ensures operational flexibility and minimizes delays. The combination of hash tables for fast parcel lookup, heaps for prioritization, and linked lists for transit management makes the system both fast and memory-efficient, handling large volumes of parcels seamlessly.

In essence, the Swift-Ex system not only simplifies the logistics management process but also enhances reliability, transparency, and operational efficiency, providing a comprehensive solution for modern logistics challenges.

## Problem Statement

Managing a logistics network involves multiple challenges that are difficult to handle manually. One of the main issues is parcel prioritization, as parcels vary in urgency, size, and handling requirements. Efficiently selecting and dispatching high-priority parcels is critical to maintaining customer satisfaction. Another challenge is route optimization, which requires calculating the most efficient delivery paths while considering real-time road conditions and traffic. Improper routing can lead to delays and increased operational costs.

Rider assignment is another complex task, as it requires matching parcels with riders based on availability, workload, and parcel requirements. Assigning riders manually is prone to errors and inefficiencies, especially in high-volume operations. The system also needs real-time monitoring, as parcels may face unexpected obstacles during transit. Without timely updates, logistics

managers cannot respond quickly to delays or route blockages. Lastly, maintaining historical tracking of parcels is essential for audits, customer inquiries, and performance analysis. Traditional manual systems often fail to provide accurate and timely data, leading to customer dissatisfaction and operational inefficiencies.

The Swift-Ex system addresses these problems by providing an automated, intelligent, and dynamic platform that prioritizes parcels, calculates efficient routes, assigns riders optimally, monitors deliveries in real-time, and maintains comprehensive historical data.

## Modules Explanation

The Parcel Management Module handles all parcel-related operations. It allows the creation of new parcels, updates their status during transit, and maintains historical tracking. The module uses hash tables for fast parcel lookup, heaps for priority-based sorting, and linked lists for managing parcels currently in transit. This ensures quick retrieval, efficient prioritization, and smooth real-time tracking.

The Rider Management Module manages rider data, including availability, current load, and type-based suitability. It implements a scoring system to assign parcels efficiently, taking into account rider capacity and parcel priority. This module automatically updates rider availability after deliveries, ensuring a balanced workload across all riders.

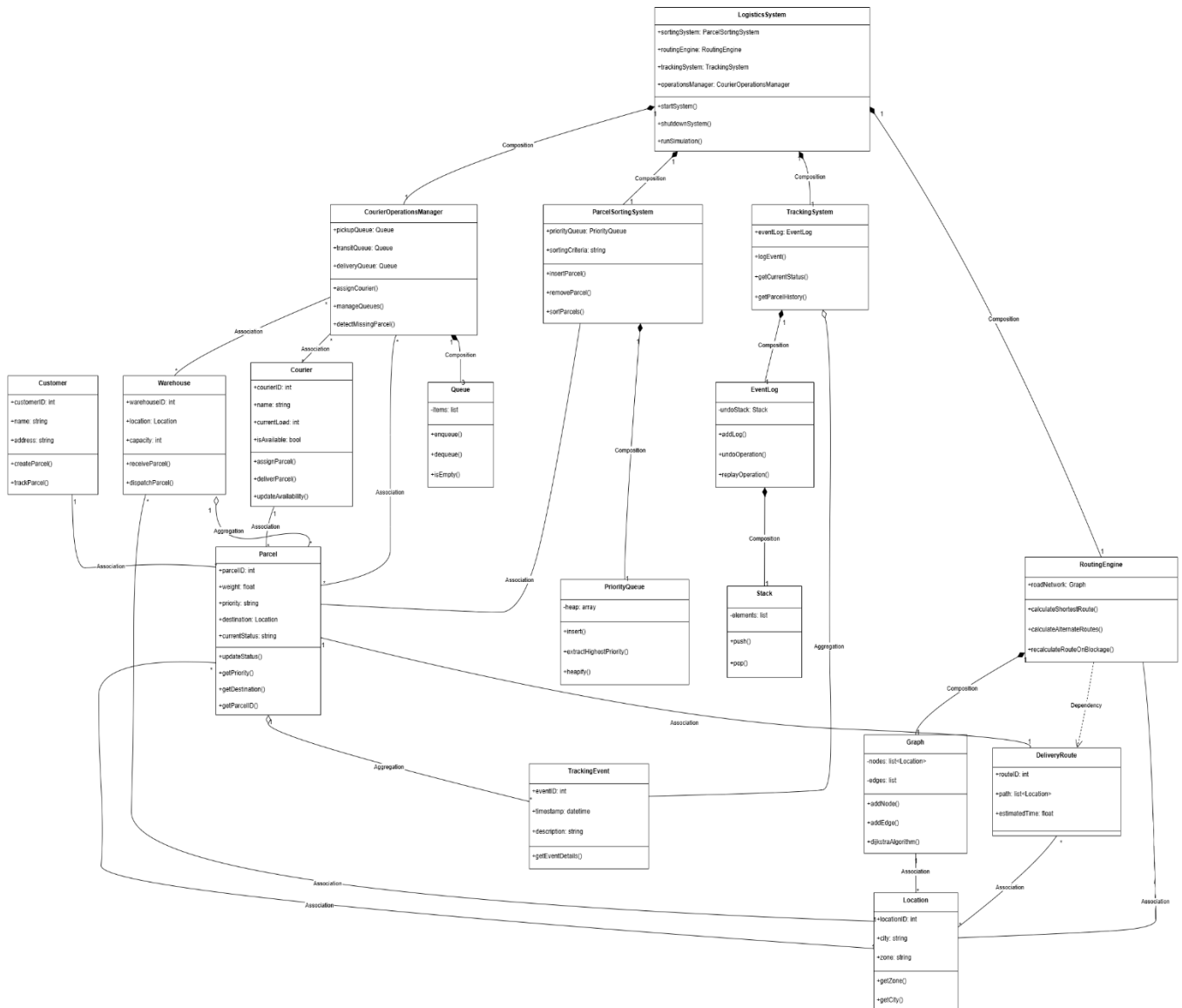
The Map & Route Management Module represents the logistics network as a graph, with cities as nodes and roads as edges. It uses Depth-First Search (DFS) to explore possible routes between source and destination and filters out blocked or overloaded paths. The module stores the top five shortest paths and allows selection based on optimal delivery time or manual preference. Real-time adjustments are made if routes become unavailable due to traffic or blockages.

The Transit Monitoring Module continuously tracks parcels in transit. It records the parcel's current location, percentage completion, and expected delivery time. Any anomalies, such as missing parcels or delayed updates, are immediately flagged. The module supports real-time route recalculation and automatically adjusts delivery schedules to minimize delays.

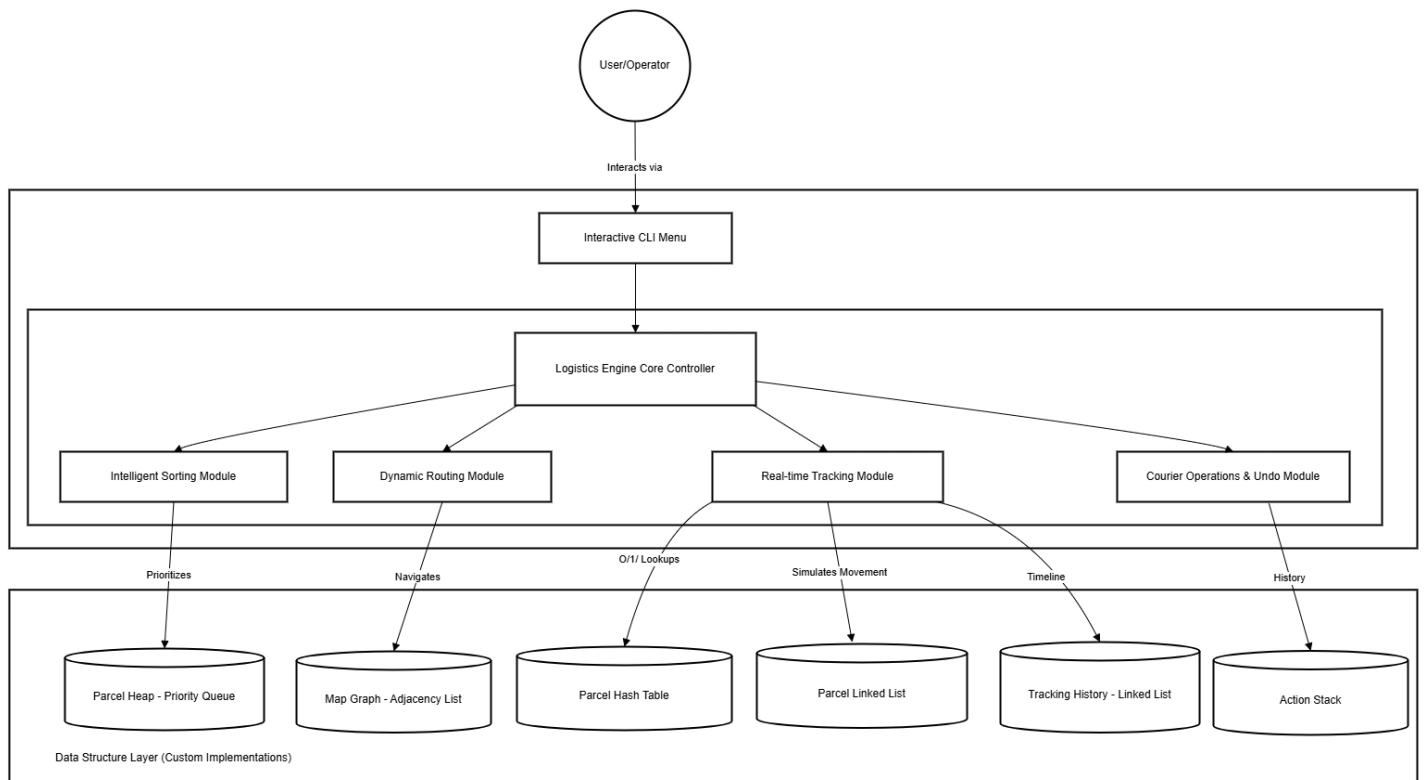
The Undo & Logging Module maintains a history of critical actions such as parcel creation, dispatch, and updates. This allows administrators to undo operations if errors occur and keeps a detailed timeline of events for each parcel, enhancing transparency and accountability.

The User Interface Module provides a clear, menu-driven interface for logistics managers. Users can add new parcels, dispatch shipments, monitor live transit, view parcel details, and manage riders. The interface ensures ease of use and clear presentation of routes, status updates, and transit progress.

## Class diagram

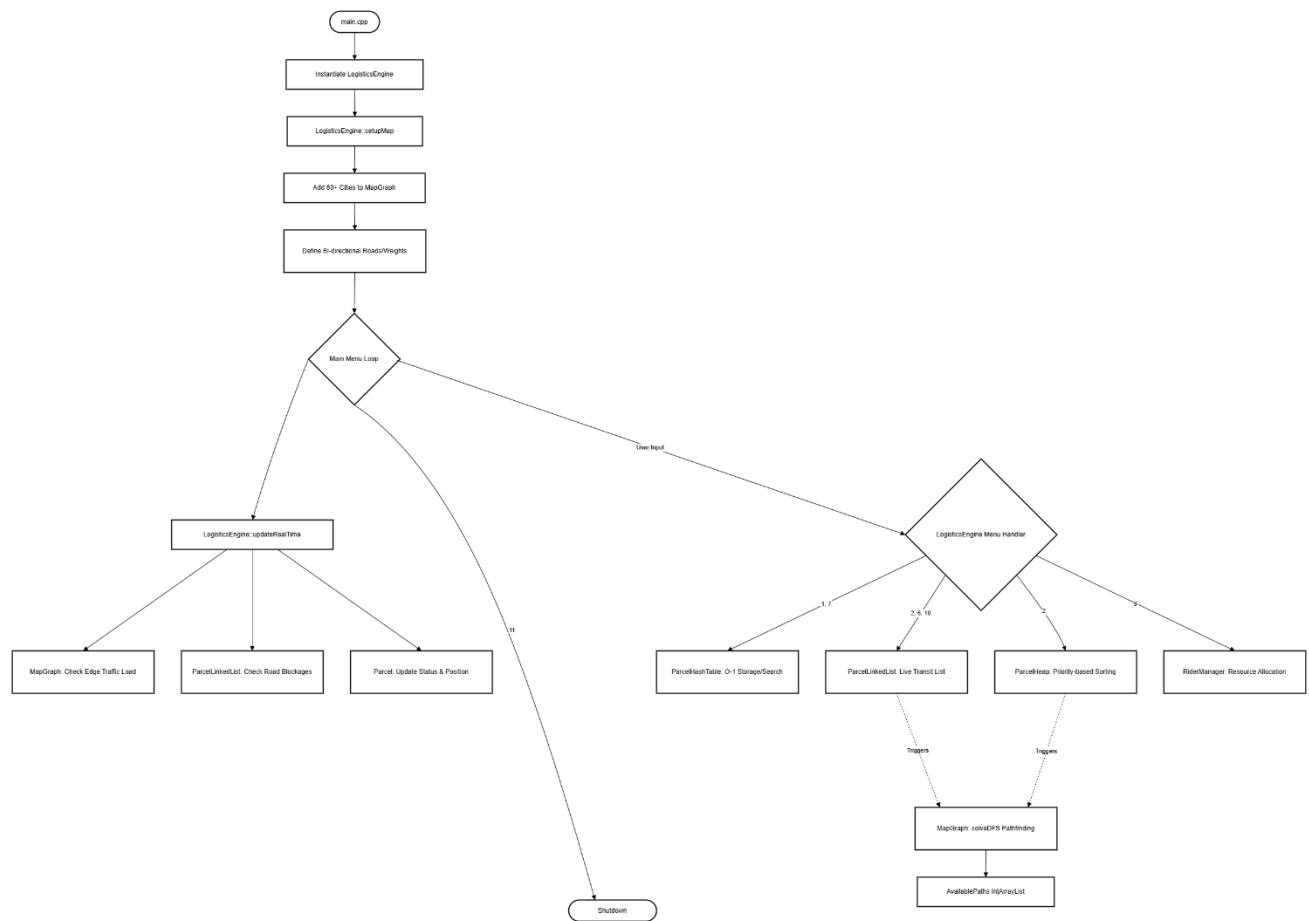


# System Architecture

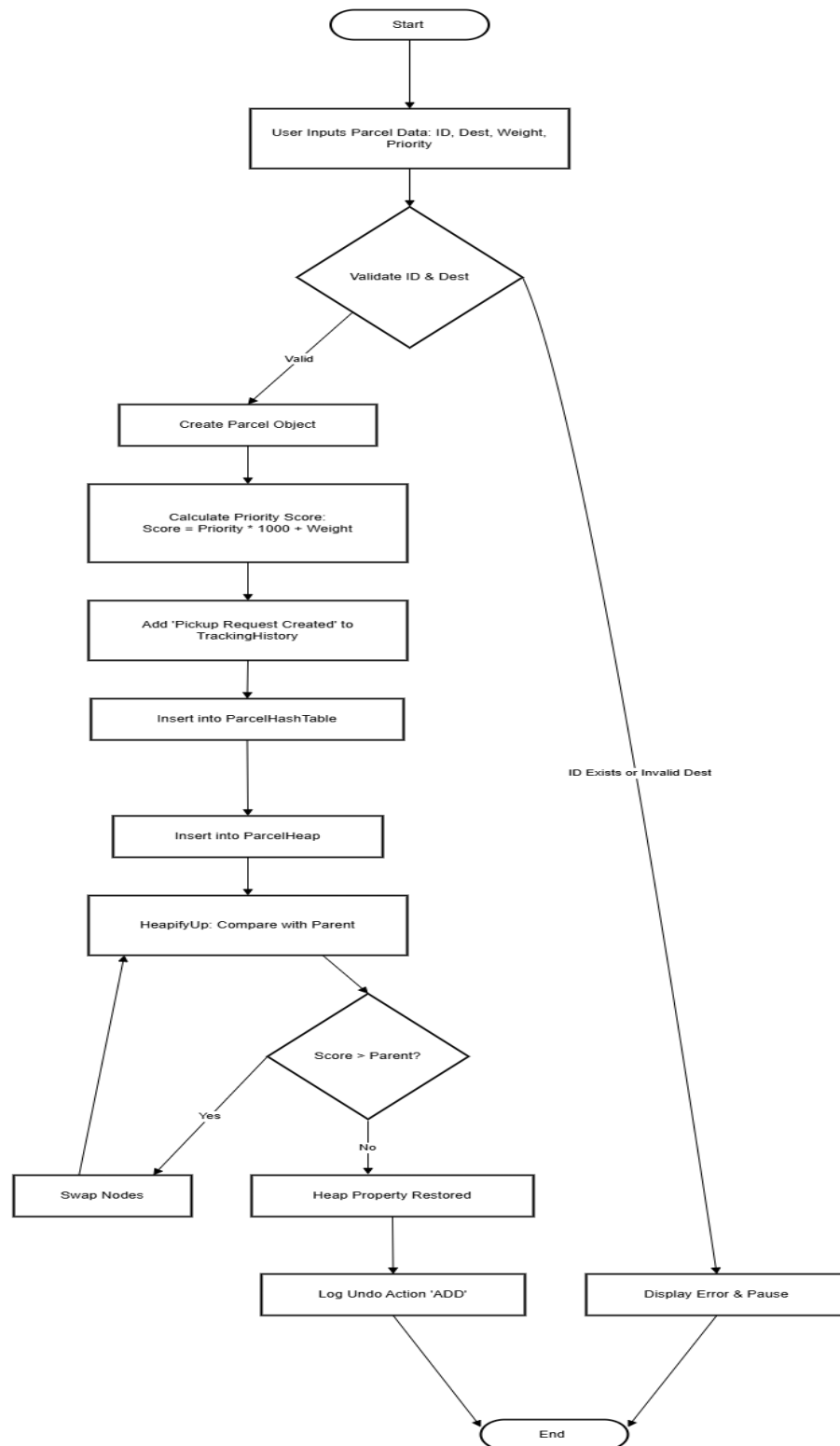


# Flowcharts

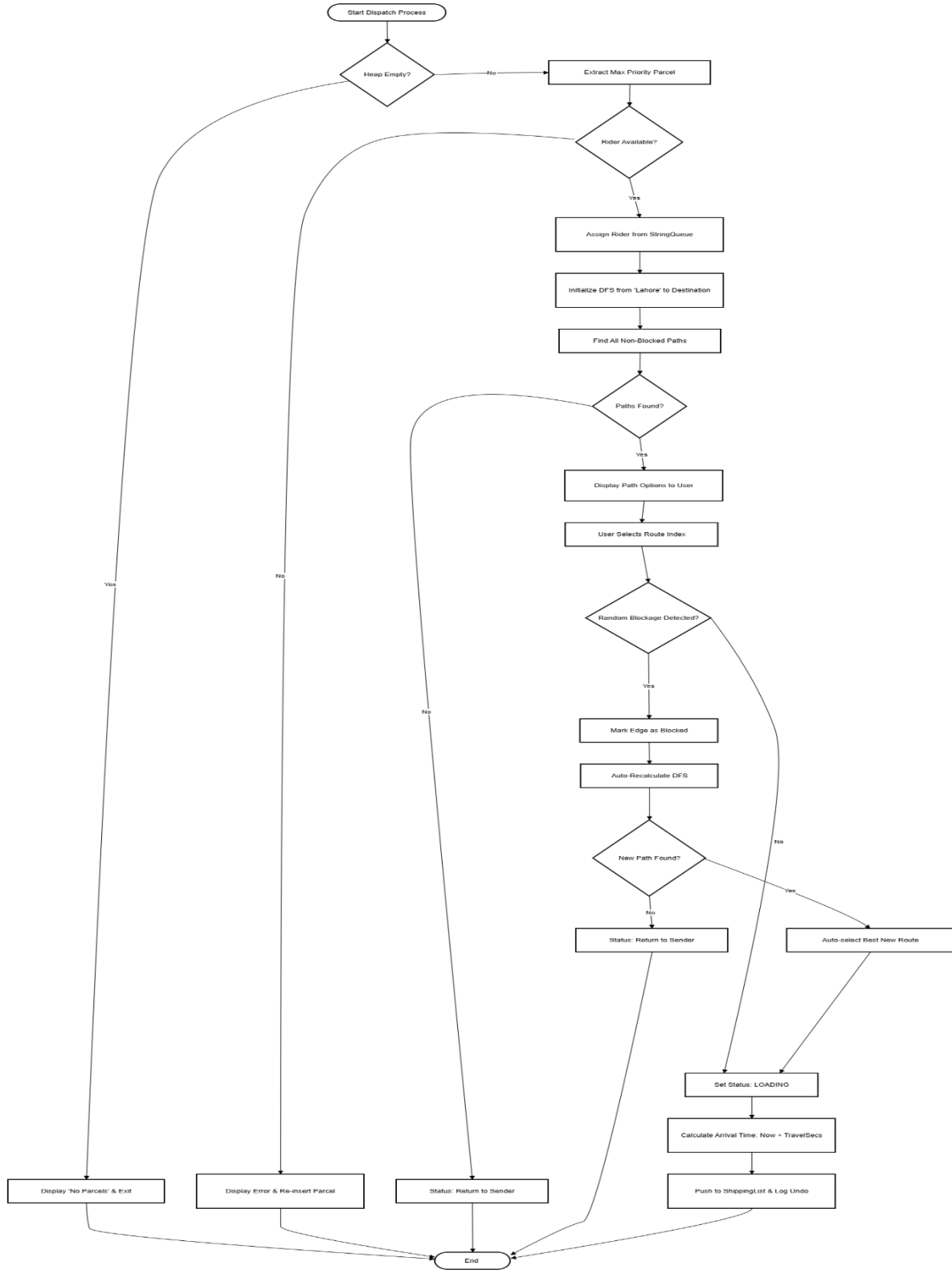
## 1. System Entry & Control Flow



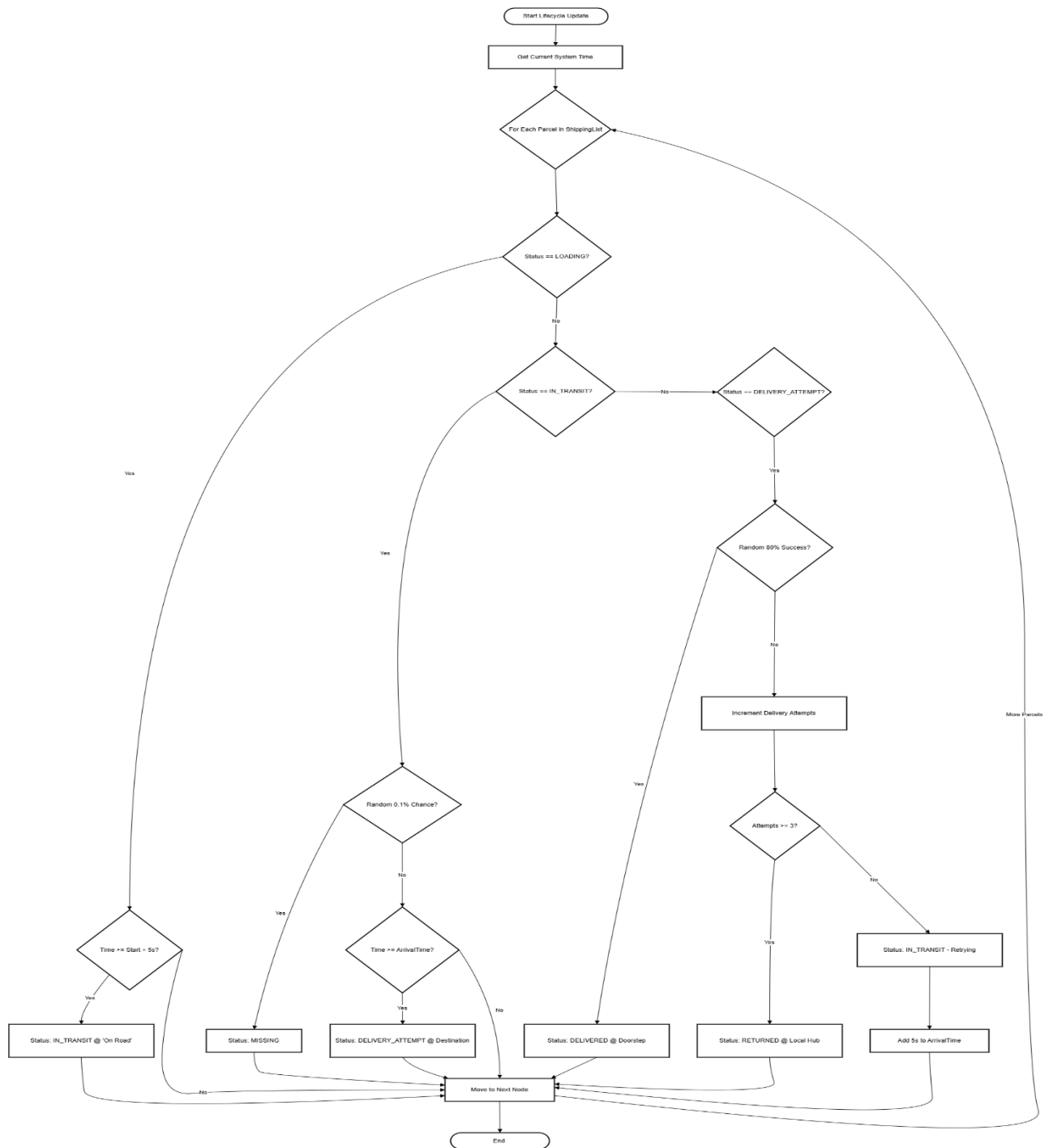
## 2. Parcel Ingestion & Priority Sorting



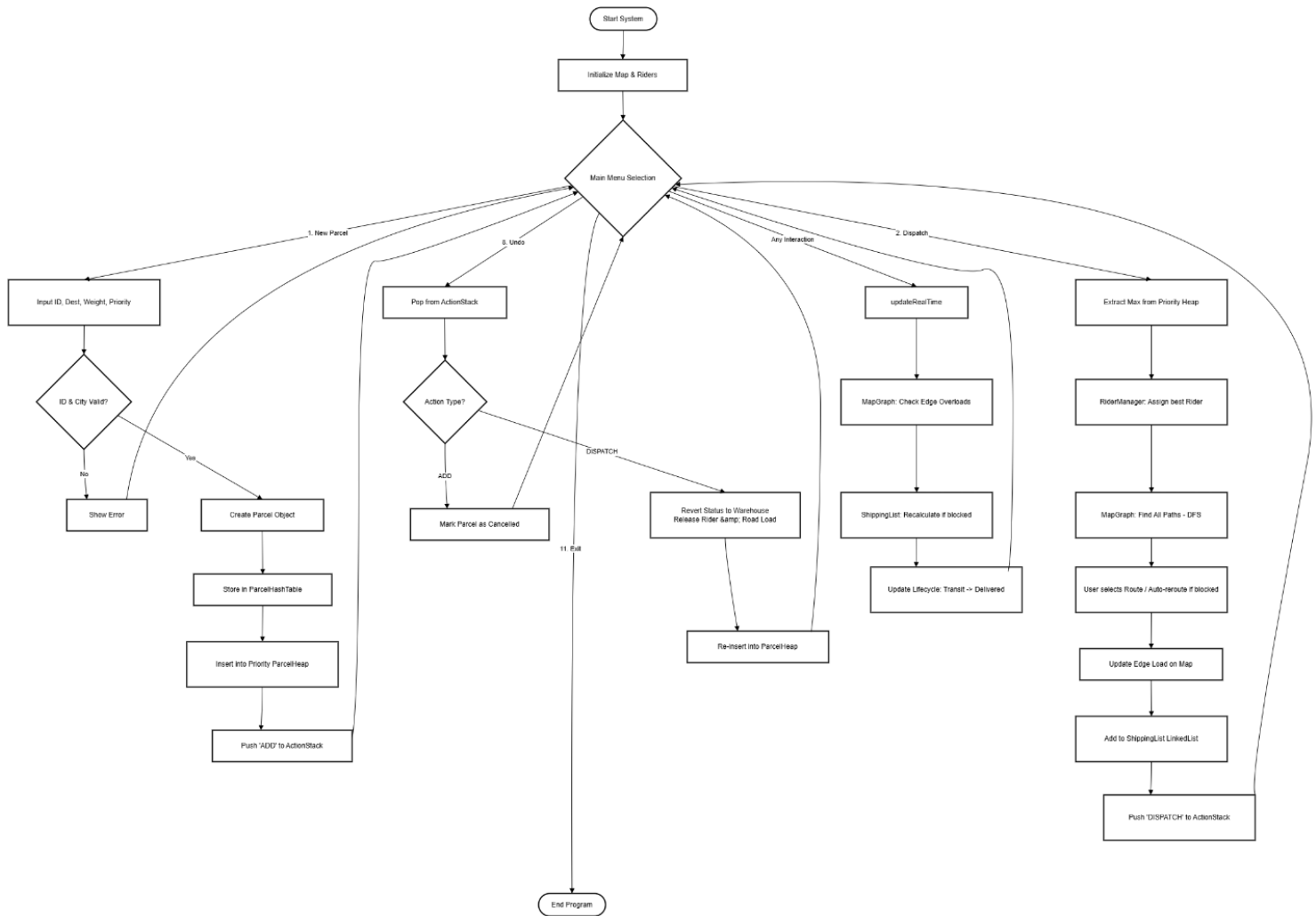
### 3. Path finding & Dispatch Logic



## 4. Real-Time Status & Lifecycle



## 5. Operations, Undo, & Persistence



## Screenshots:

Main Menu (consists of four modules):

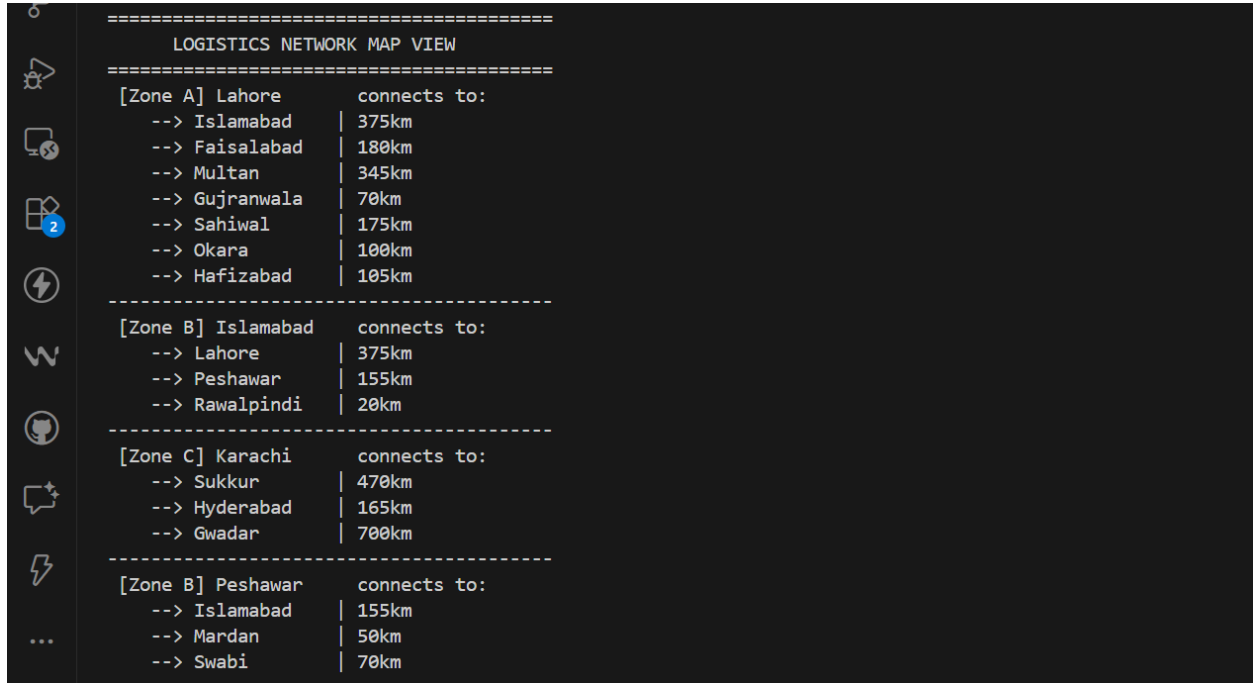
```
○  
  
===== SWIFT-EX MAIN CONTROL PANEL =====  
= 1. Intelligent Parcel Sorting Module =  
= 2. Parcel Routing Module =  
= 3. Parcel Tracking System =  
= 4. Courier Operations Engine =  
= 5. Exit System =  
=====
```

Select Module: █

Parcel Sorting Module:

```
--- INTELLIGENT PARCEL SORTING MODULE ---  
1. New Pickup Request  
2. Cancel/Withdraw Parcel  
3. View Sorting Database  
4. Return to Main Menu  
Enter choice: █
```

## Mapping:



```
=====
LOGISTICS NETWORK MAP VIEW
=====
```

[Zone A] Lahore	connects to:
--> Islamabad	375km
--> Faisalabad	180km
--> Multan	345km
--> Gujranwala	70km
--> Sahiwal	175km
--> Okara	100km
--> Hafizabad	105km

```
-----
```

[Zone B] Islamabad	connects to:
--> Lahore	375km
--> Peshawar	155km
--> Rawalpindi	20km

```
-----
```

[Zone C] Karachi	connects to:
--> Sukkur	470km
--> Hyderabad	165km
--> Gwadar	700km

```
-----
```

[Zone B] Peshawar	connects to:
--> Islamabad	155km
--> Mardan	50km
--> Swabi	70km

## Courier Operations Engine:

```
--- COURIER OPERATIONS ENGINE ---
1. Warehouse Dispatch (Process Next)
2. View Rider Status
3. Undo Last Operation
4. Return to Main Menu
Enter choice: █
```

## View Rider Status:

```
=== RIDER STATUS ===
```

Name	Type	Load	Max	Parcels	Status
Ali Khan	Light	3	50	1	Available
Bilal Ahmed	Heavy	0	200	0	Available
Usman Tariq	Priority	0	100	0	Available
Fahad Mustafa	General	0	150	0	Available

```
Press Enter to continue...
```

## Algorithm Choice & Time Complexity

The system uses heaps for parcel prioritization, providing  $O(\log n)$  time for insertion and extraction. This allows high-priority parcels to be efficiently selected over large datasets. Hash tables are used for parcel storage and retrieval, achieving  $O(1)$  average access time, making the system highly efficient for searching and updating parcels.

DFS is employed for route calculation, enabling the system to explore all possible paths between source and destination. It is chosen over Dijkstra's algorithm because the system needs multiple shortest paths rather than just the single shortest path. Linked lists are used for transit tracking, allowing easy insertion, deletion, and traversal of parcels currently in transit. These algorithm choices ensure that the system can handle hundreds of parcels, dynamically respond to changes, and maintain optimal performance.

## Drawbacks

Despite its efficiency, the system has certain limitations. The DFS-based route calculation can become computationally expensive in very large networks with many cities and roads. The system does not currently support real-time integration with external traffic APIs, which limits its ability to react to real-world traffic conditions instantly. While heaps and hash tables provide fast access and prioritization, they require careful memory management, which can be challenging in extremely high-volume logistics operations. Additionally, the user interface is console-based and may not provide the advanced visualization or reporting features that some users may require.

## Links

### GitHub Link

<https://github.com/ahmar2006/SwiftEx-Logistics-System.git>

### Linkedin Link

[https://www.linkedin.com/posts/muhammad-fahad-9b9639331\\_excited-to-share-our-group-project-swift-ex-activity-7412213218665754624-kmHb?utm\\_source=share&utm\\_medium=member\\_desktop&rcm=ACoAAF7-6q4BmDStSTLyA2uAJrmkRNUVexDNLXQ](https://www.linkedin.com/posts/muhammad-fahad-9b9639331_excited-to-share-our-group-project-swift-ex-activity-7412213218665754624-kmHb?utm_source=share&utm_medium=member_desktop&rcm=ACoAAF7-6q4BmDStSTLyA2uAJrmkRNUVexDNLXQ)

## Conclusion

The Swift-Ex Intelligent Logistics System is a comprehensive, modular, and efficient solution for modern parcel delivery management. It automates key operations, including parcel registration, route optimization, rider assignment, and real-time transit monitoring. The system's use of intelligent algorithms and efficient data structures ensures high performance, even with large volumes of parcels. While certain limitations exist, such as computational overhead for very large networks and the lack of advanced visualization features, the system significantly improves delivery efficiency, reduces human error, and provides comprehensive tracking and accountability. Overall, Swift-Ex demonstrates how automation and intelligent algorithms can transform logistics management into a faster, more reliable, and customer-friendly operation. Video link is given below: