TEAM NAME : TEAM LOGISTICS
DOMAIN NAME : LOGISTICS AND DELIVERY OPERATIONS
PROBLEM STATEMENT : FLEET AND DRIVER OPERATION MANAGEMENT SYSTEM

Submitted by:

STUDENT NAME	ROLL NUMBER
K SRUJAN	S20240010116
T GRANTHIK	S20240010240
K TARUN	S20240010100
P KOUSHIK	S20240010174
D HARSHAVEER	S20240010061

Problem Statement Explanation

The Fleet and Driver Operations Management System is developed to improve and automate logistics and delivery operations. It helps organizations manage fleet vehicles, drivers, shipments, and delivery processes through a single centralized platform.

In many logistics companies, activities like assigning drivers, tracking vehicles, preparing shipments, monitoring compliance, and confirming deliveries are handled manually or using multiple disconnected systems. This often results in delays, poor coordination, lack of real-time visibility, and higher chances of errors.

This system solves these problems by integrating all logistics operations into one system. It allows better fleet and driver management, supports automated delivery planning and dispatch, enables real-time delivery tracking, ensures compliance validation, improves warehouse coordination, and allows customers to track and confirm deliveries easily. It also helps handle delivery exceptions like delays, breakdowns, or reassignment efficiently.

The main goal of the system is to create a smooth end-to-end delivery workflow from shipment preparation to final delivery confirmation while improving operational efficiency and reducing manual effort.

Identified Actors

The system involves the following actors :

- Admin
- Fleet Manager
- Dispatcher
- Driver
- Warehouse Operator
- Customer

USER REQUIREMENTS

Admin

- Manage users and roles
- Configure system rules like working hours and SLA limits
- Manage master data such as vehicle types, zones, and package types
- Monitor system access and security

Fleet Manager

- Add, update, allocate, or block vehicles
- Track vehicle maintenance schedules
- Monitor driver license and document validity
- Track fuel usage and vehicle mileage
- View fleet performance analytics

Dispatcher

- Create delivery orders
- Assign drivers and vehicles
- Plan delivery routes
- Reassign deliveries during issues
- Handle delivery exceptions

Warehouse Operator

- Prepare shipments (picking, packing, labeling, verification)
- Mark shipment ready for dispatch
- Handover packages to drivers
- Report shipment issues

Driver

- Check in and check out shifts
- Accept or reject delivery tasks
- Update delivery status
- Submit proof of delivery (OTP, signature, photo)
- Report delivery issues

Customer

- Track delivery status
- Confirm delivery using OTP
- Provide delivery feedback

SYSTEM REQUIREMENTS

- Provide centralized logistics management system
- Store fleet, driver, and delivery data
- Support real time tracking
- Provide secure login and access control
- Maintain delivery history and logs
- Support high system availability
- Support system scalability

FUNCTIONAL REQUIREMENTS:

A Functional Requirement describes what the system should do. It specifies the features,actions, or services the system must provide to User.

- User login and role management
- Fleet and vehicle management
- Driver compliance checking
- Delivery order creation and assignment
- Route planning and optimization
- Shipment status updates
- Real time delivery tracking
- Proof of delivery submission
- Delivery exception handling
- Customer tracking and feedback system

Non Functional Requirements :

A Non-Functional Requirement describes how well the system should perform.
It specifies quality attributes such as performance, security, reliability, usability, and scalability.

Performance

- Should handle multiple deliveries at same time
- Tracking updates should be near real time

Security

- Role based access control
- Secure data storage
- Secure data transmission

Reliability

- High system uptime
- No data loss during failures

Usability

- Easy to use interface
- Mobile friendly for drivers

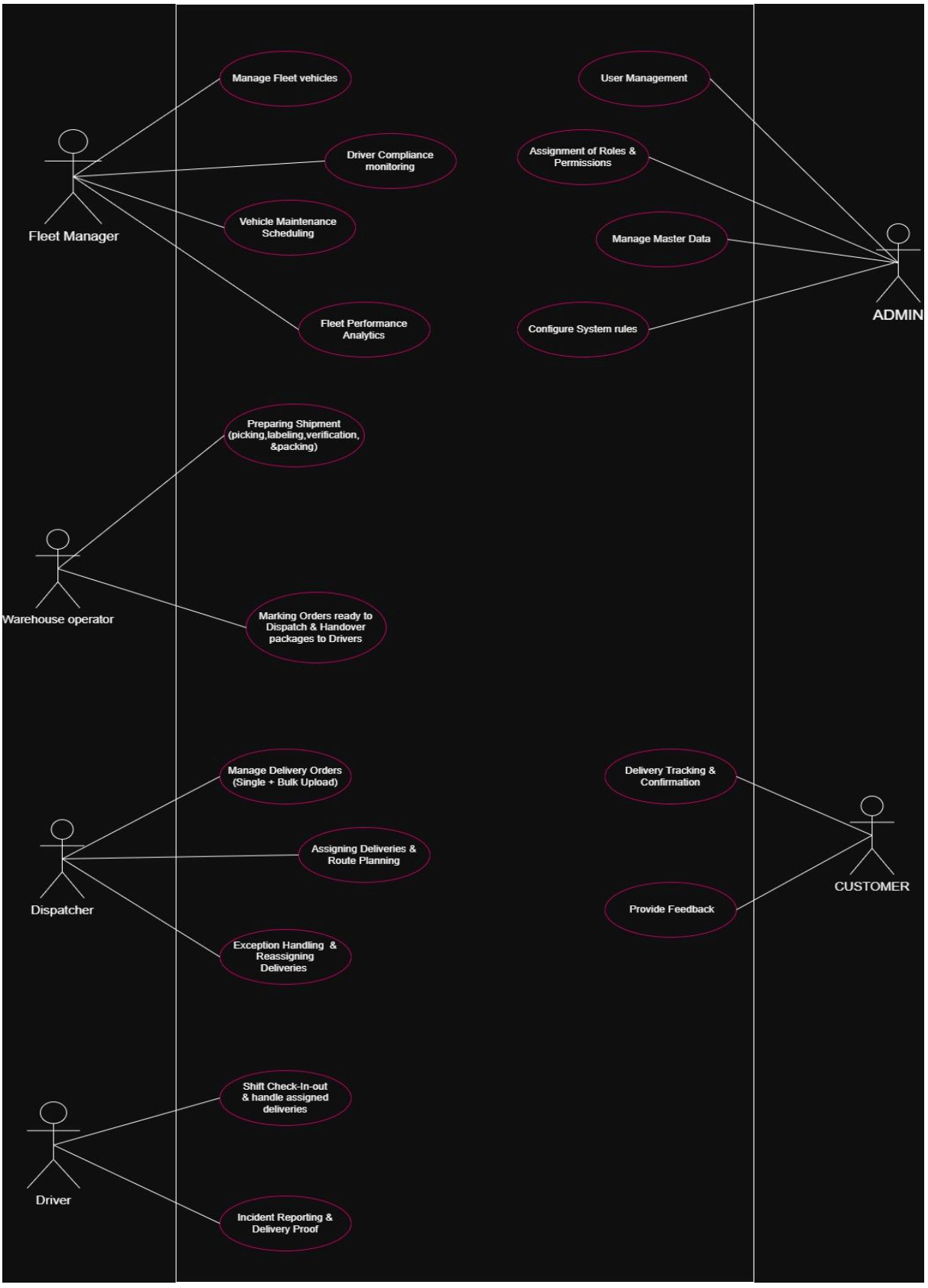
Scalability

- Should support increase in users, vehicles, and deliveries

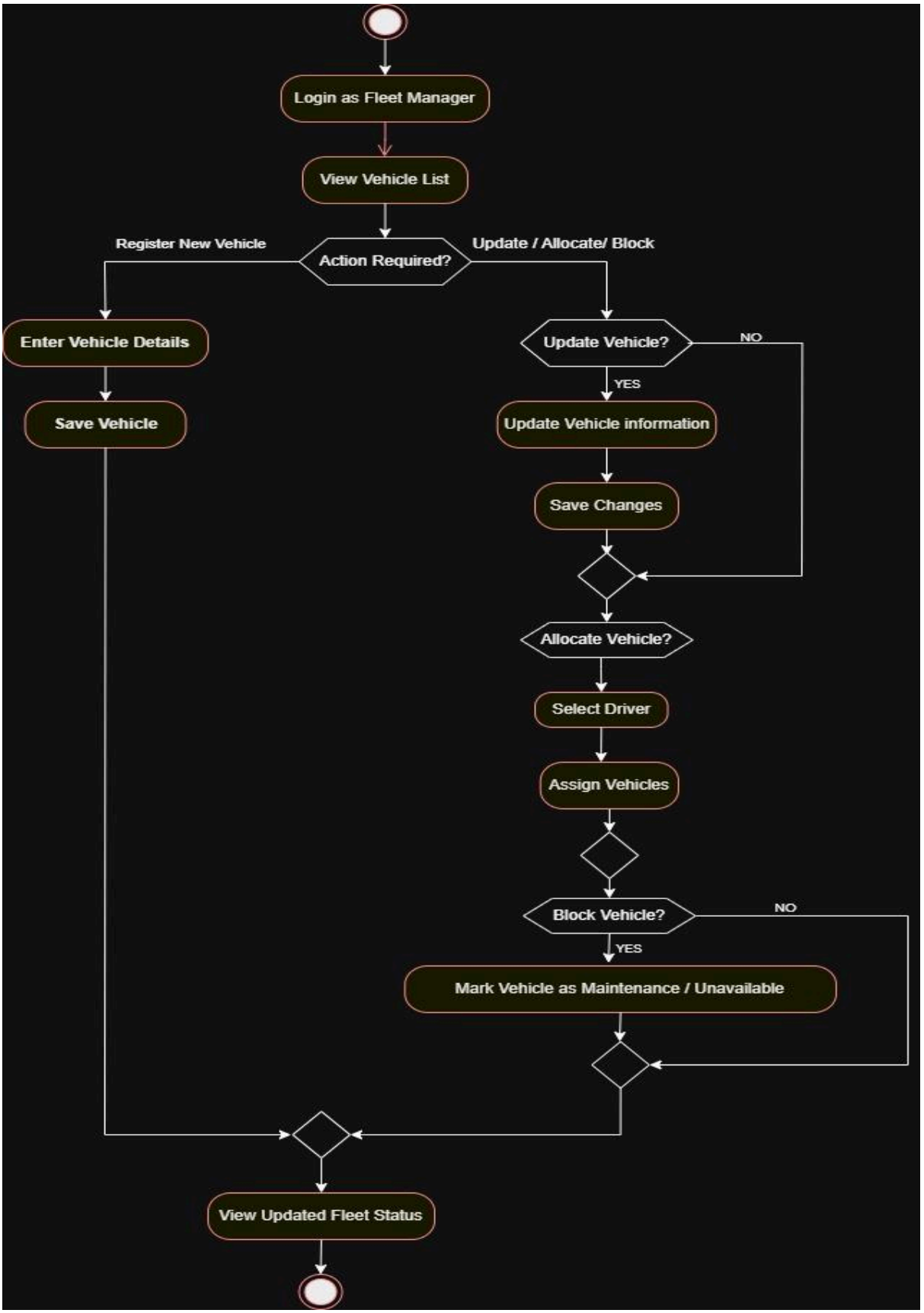
UML DIAGRAMS(USE CASE, ACTIVITY, SEQUENCE DIAGRAM) -

GOOGLE DRIVE LINK -
https://drive.google.com/file/d/1xpYuYgWomhrW1L3O7MptQOLPz_3Mq4Gx/view?usp=sharing

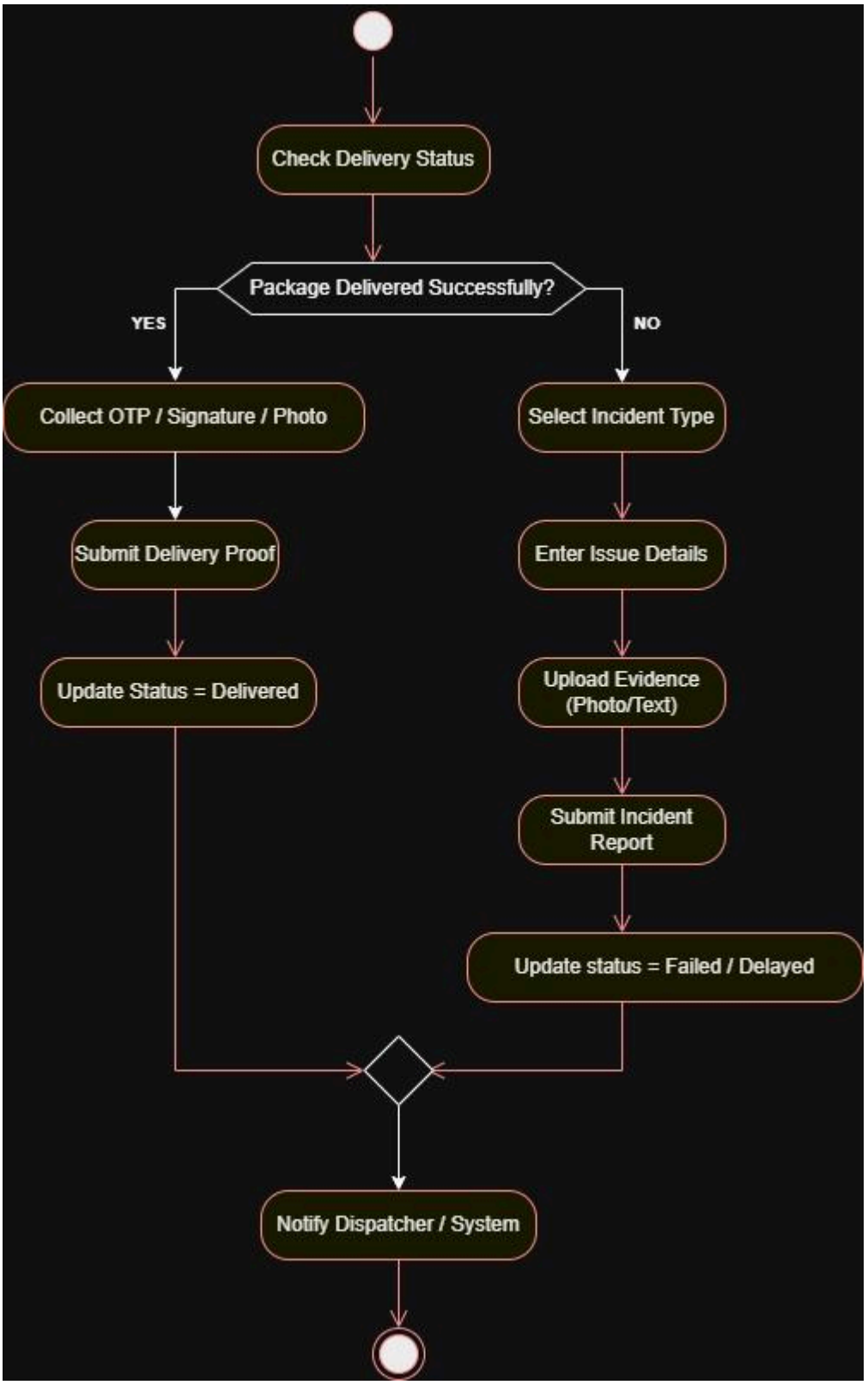
USE CASE DIAGRAM:



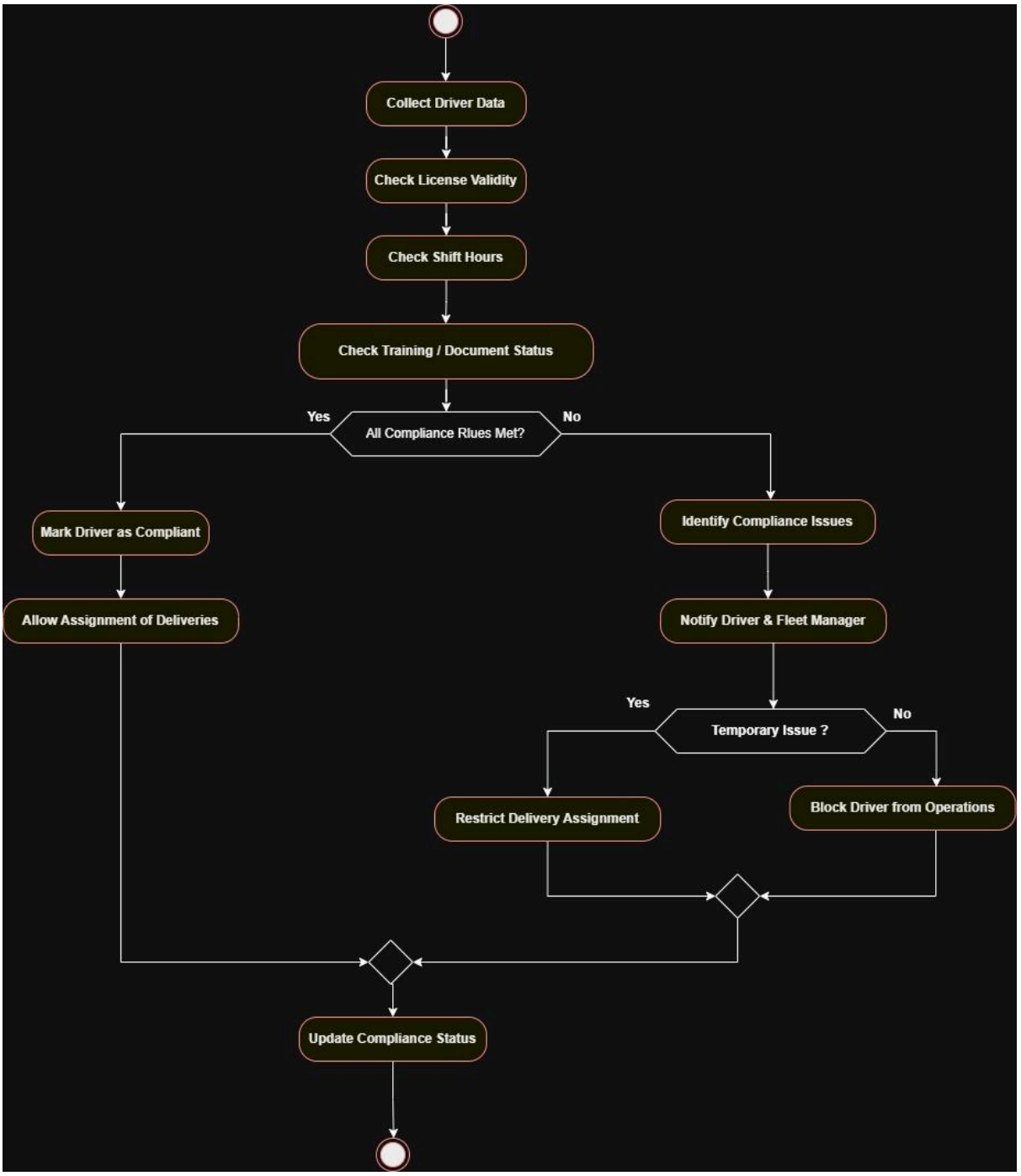
ACTIVITY DIAGRAMS :
MANAGE FLEET VEHICLES-



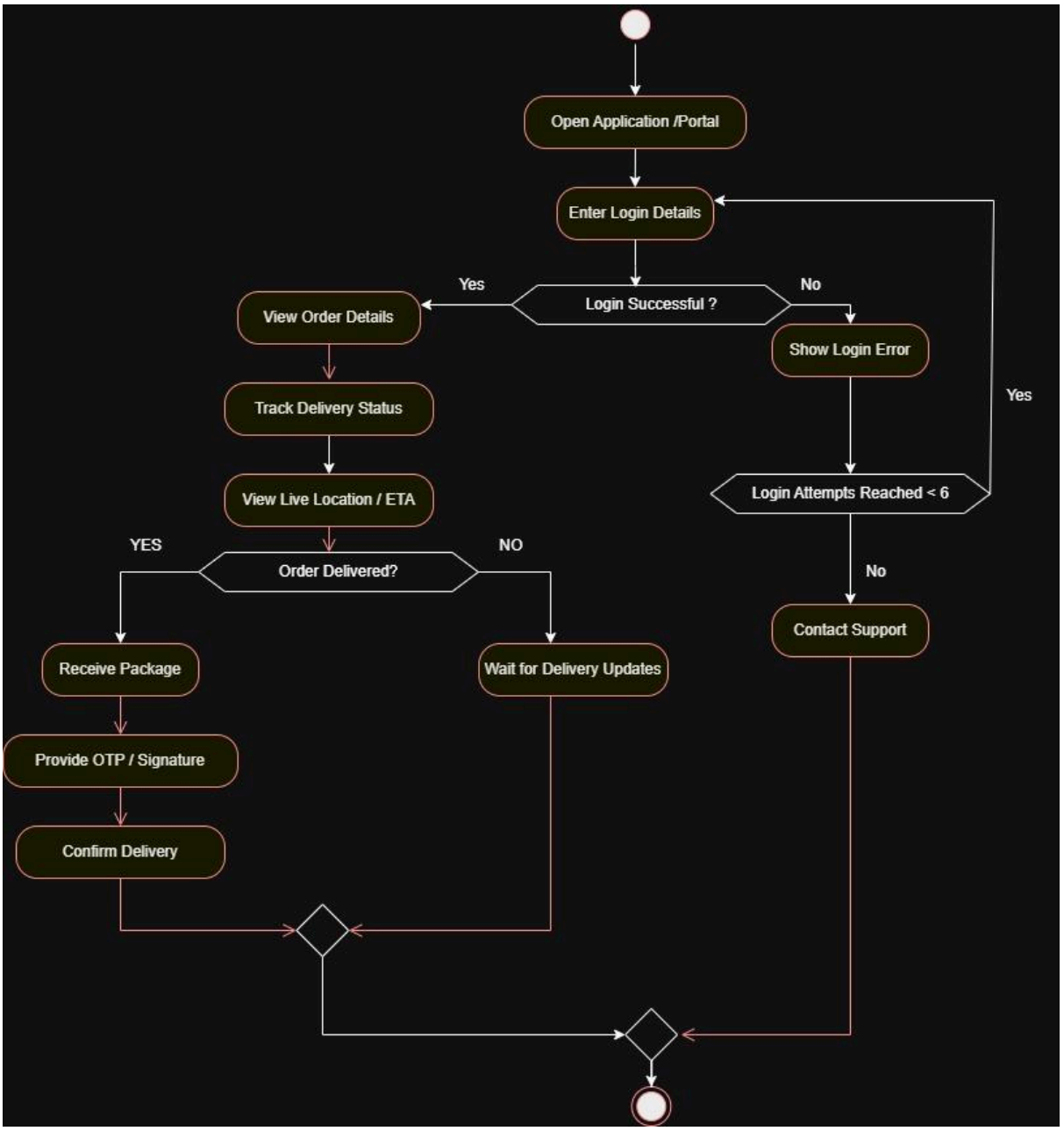
INCIDENT REPORTING AND DELIVERY PROOF-



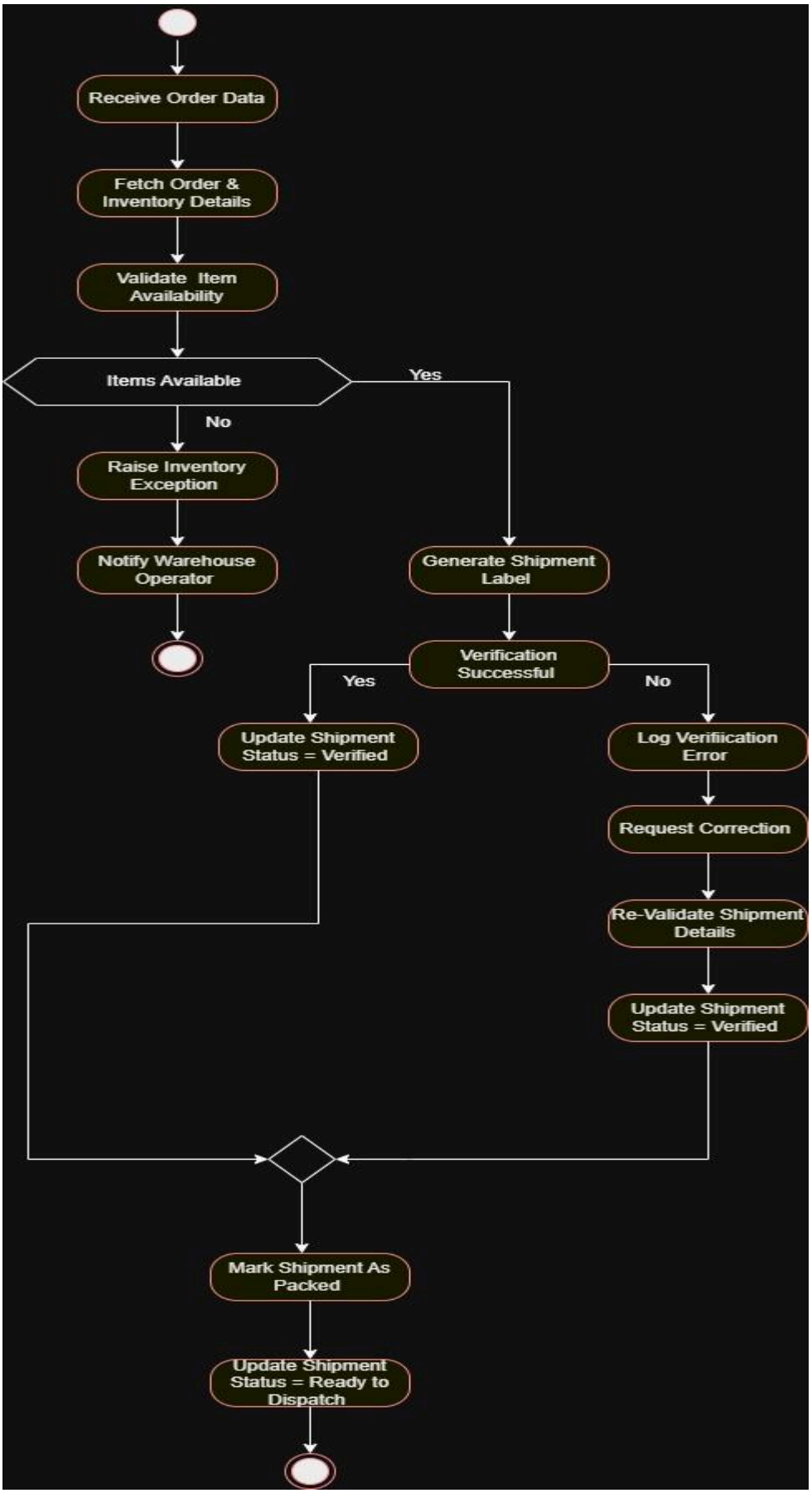
DRIVER COMPLIANCE MONITORING -



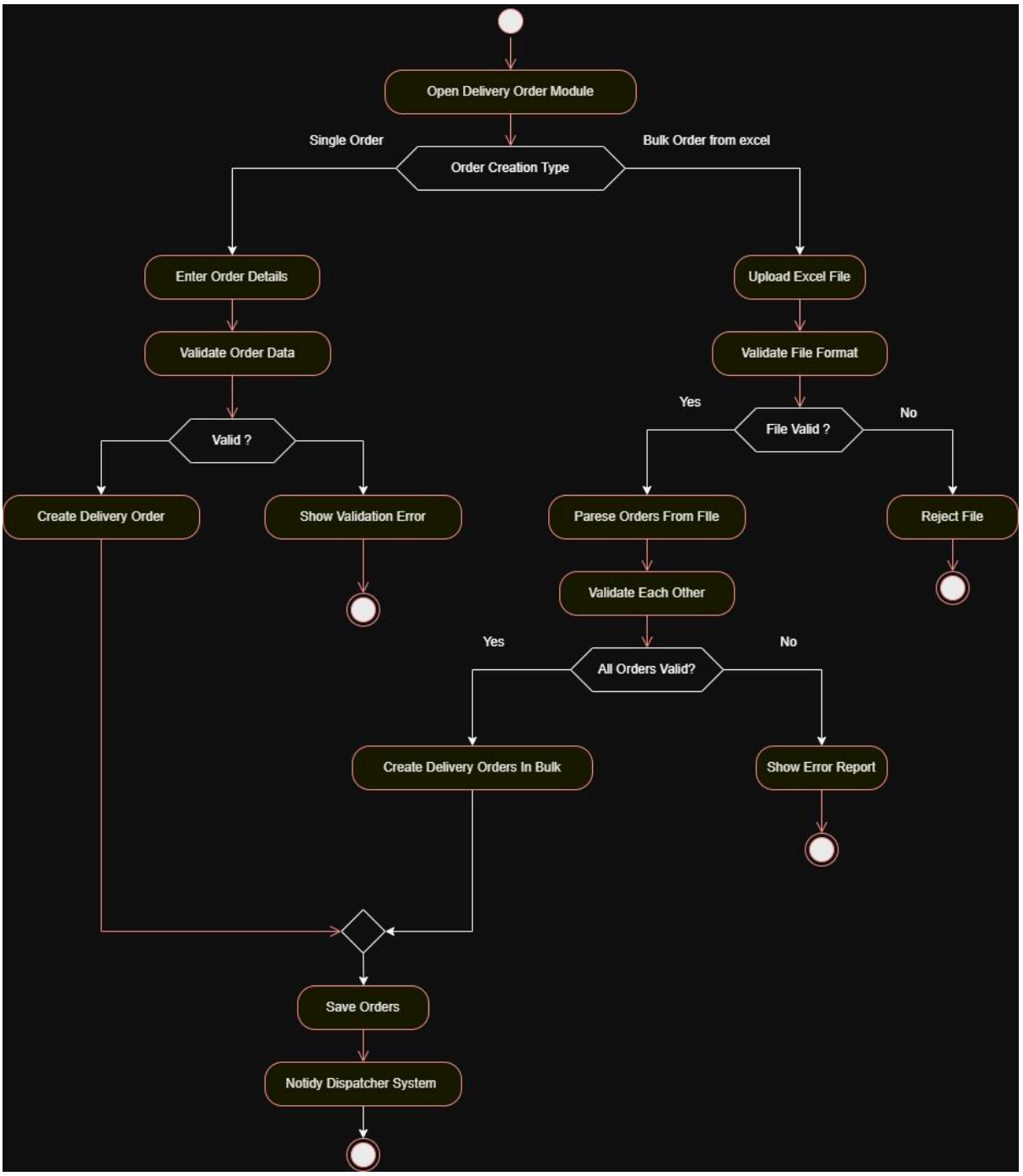
DELIVERY TRACKING AND CONFIRMATION-



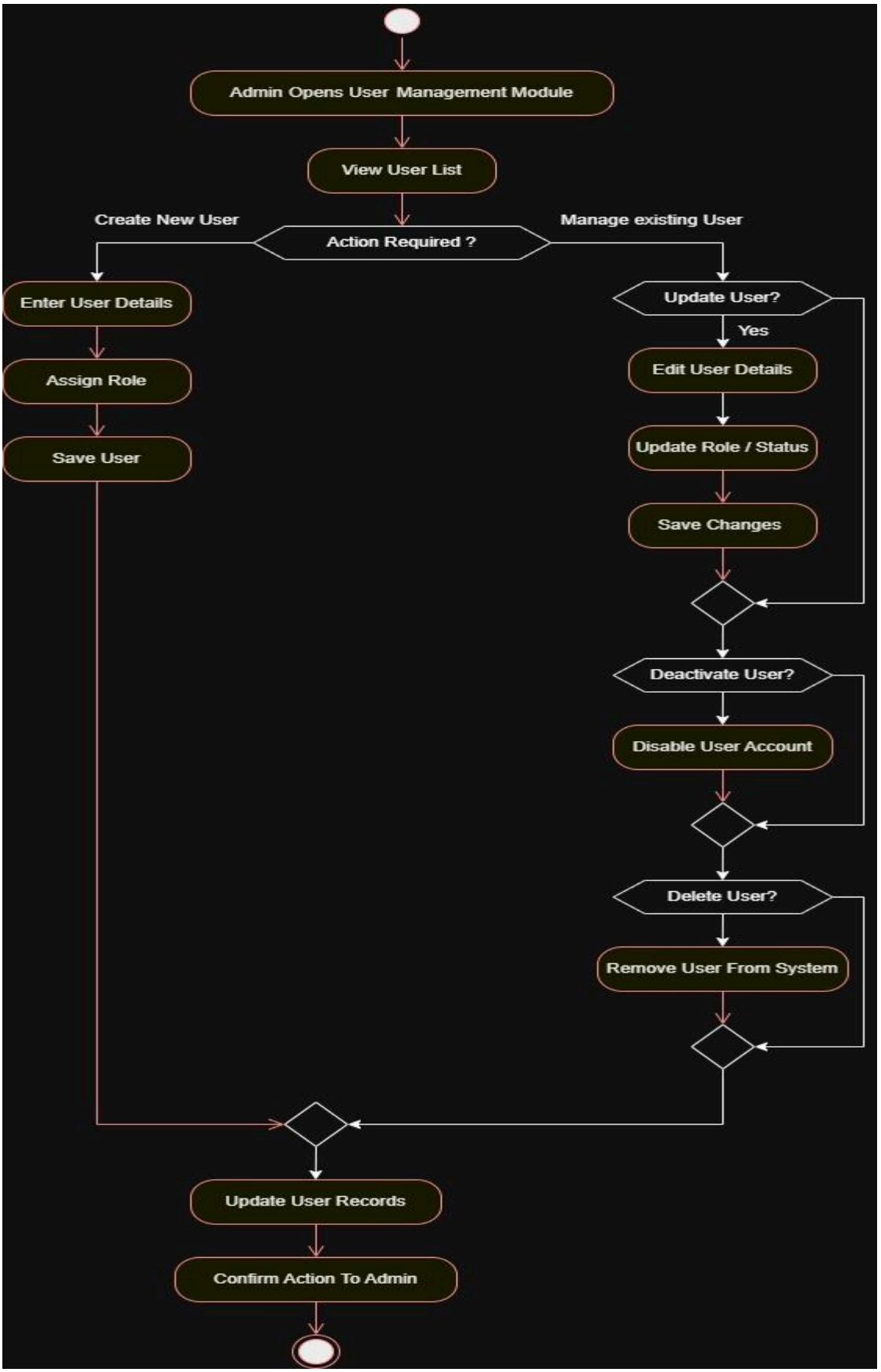
PREPARING SHIPMENT -



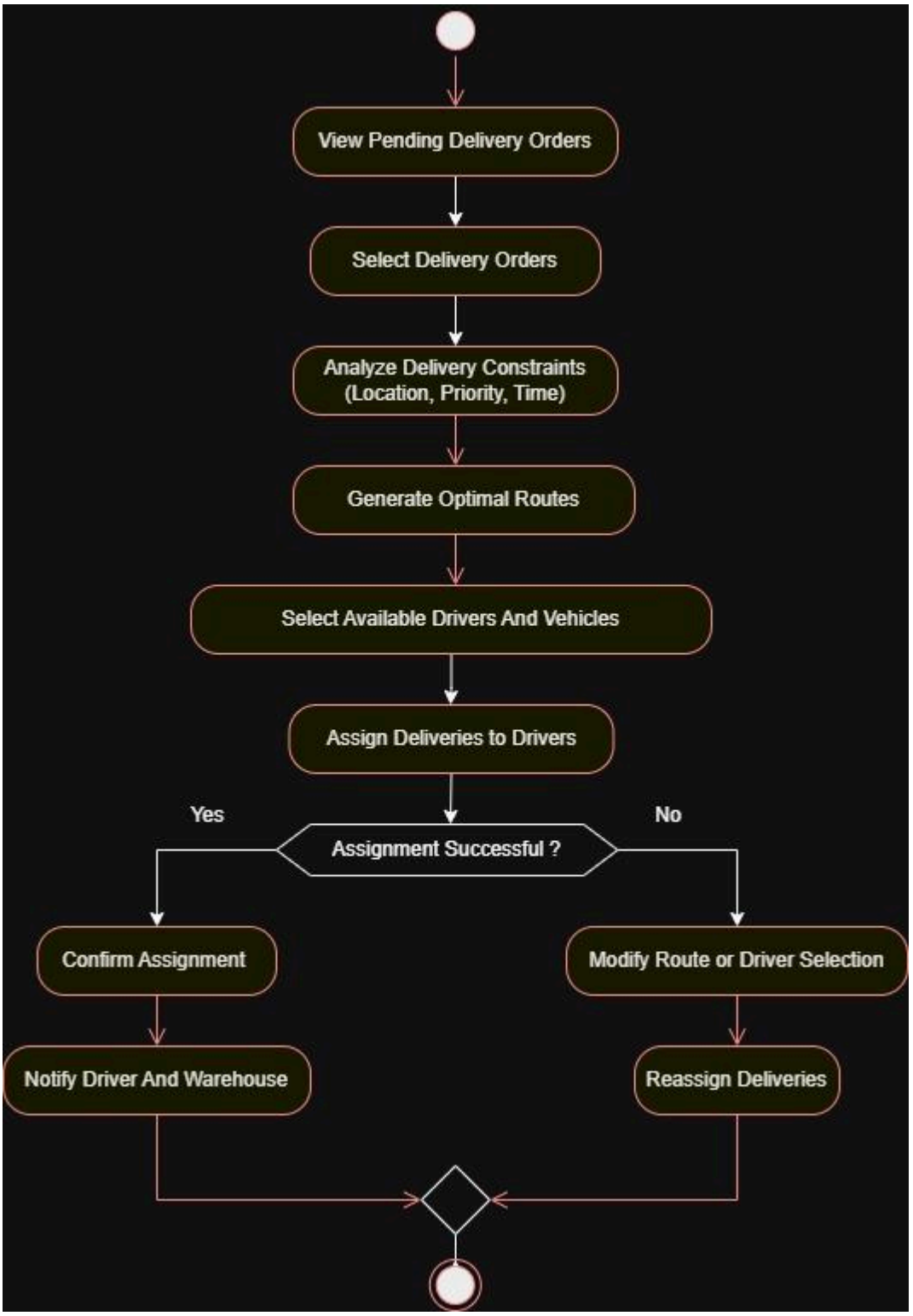
MANAGE DELIVERY ORDERS -



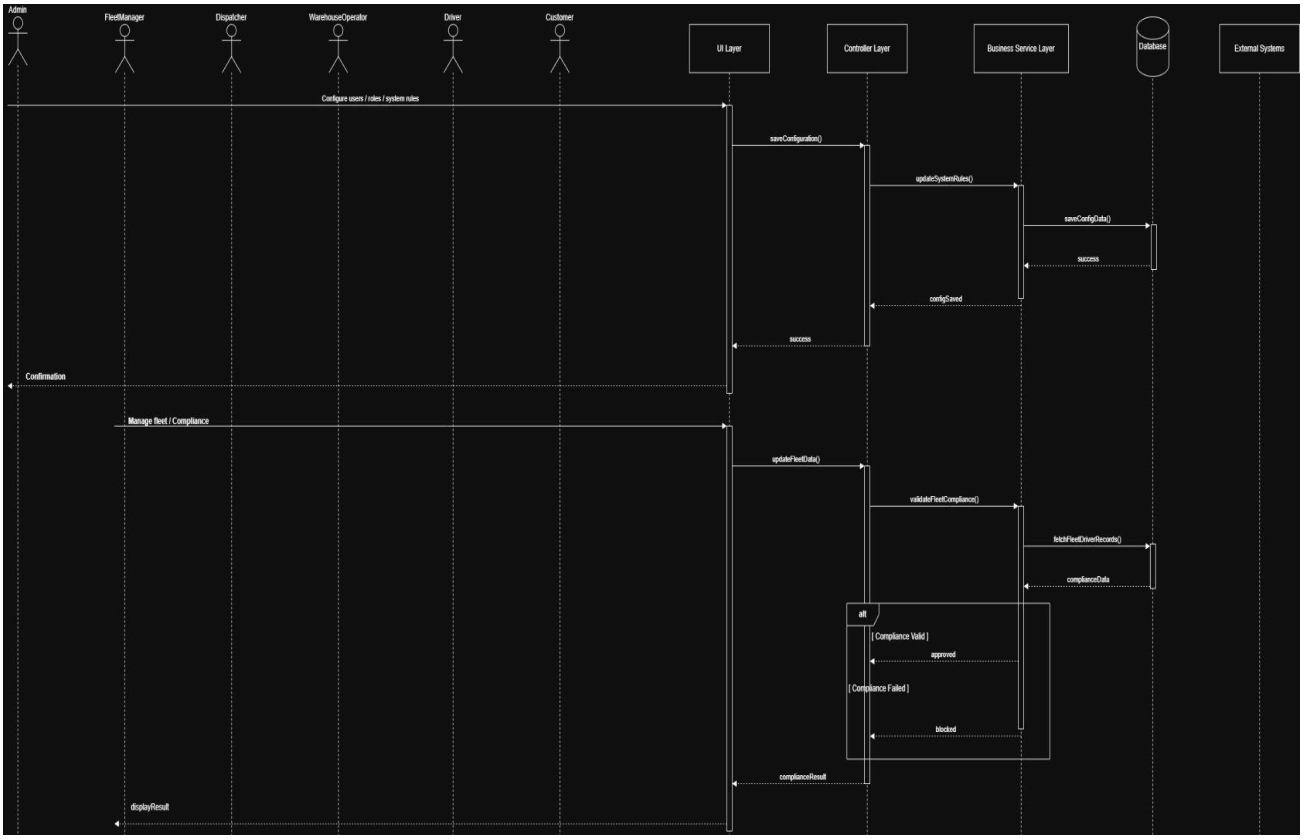
USER MANAGEMENT -

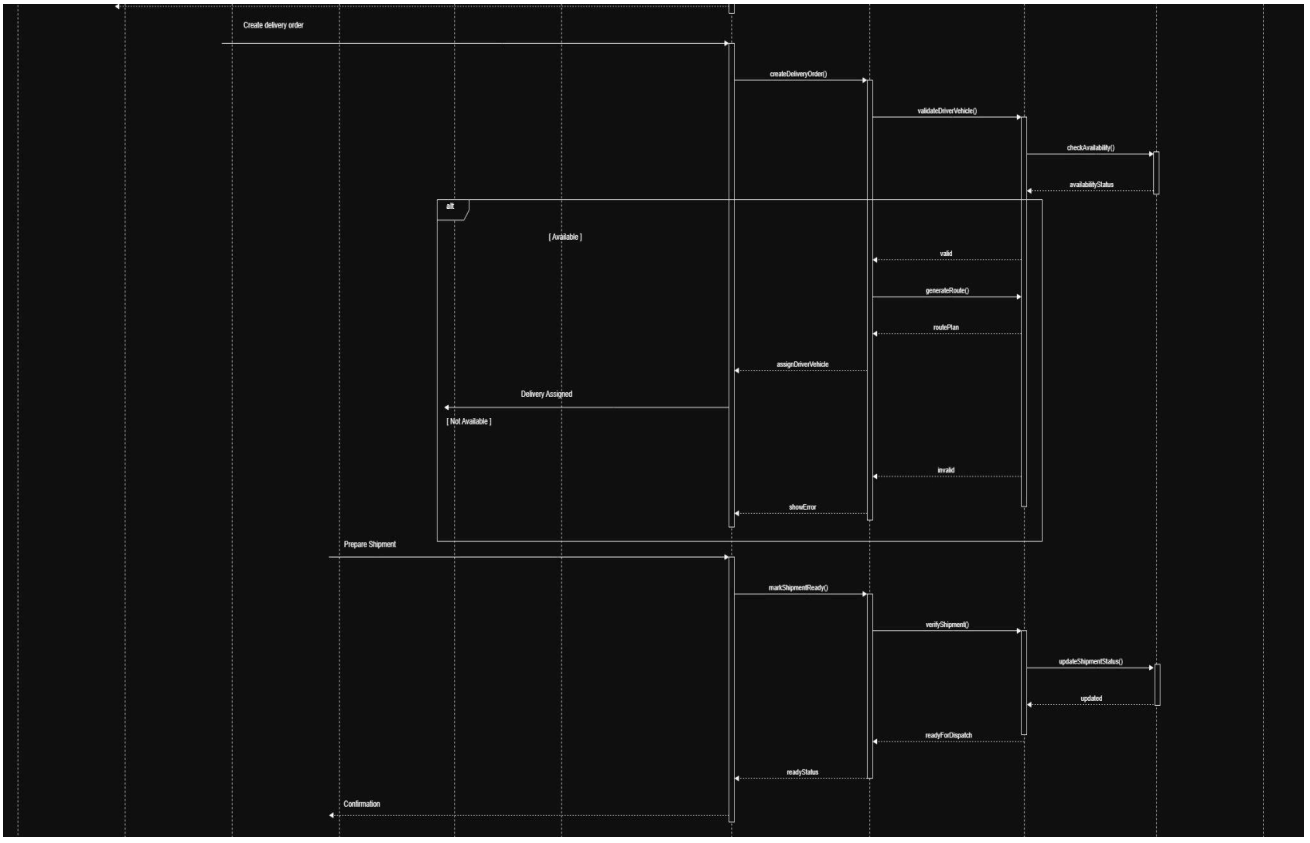


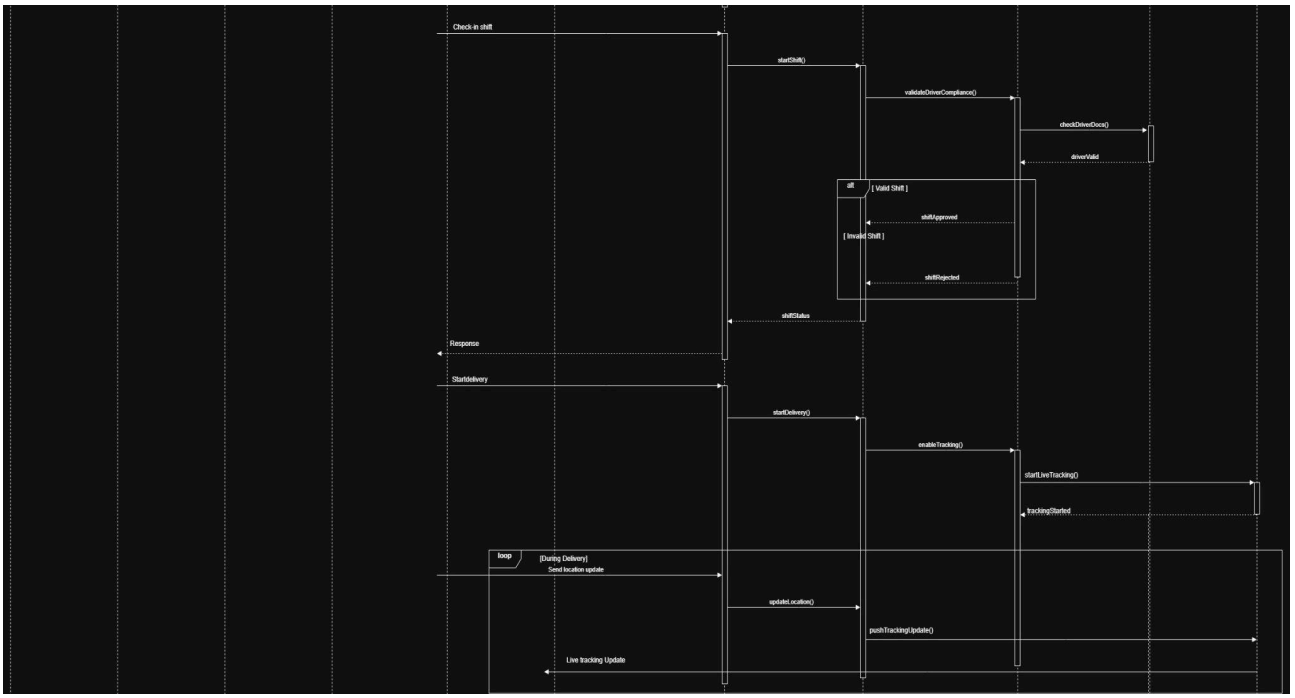
ASSIGNING DELIVERIES AND ROUTE PLANNING-

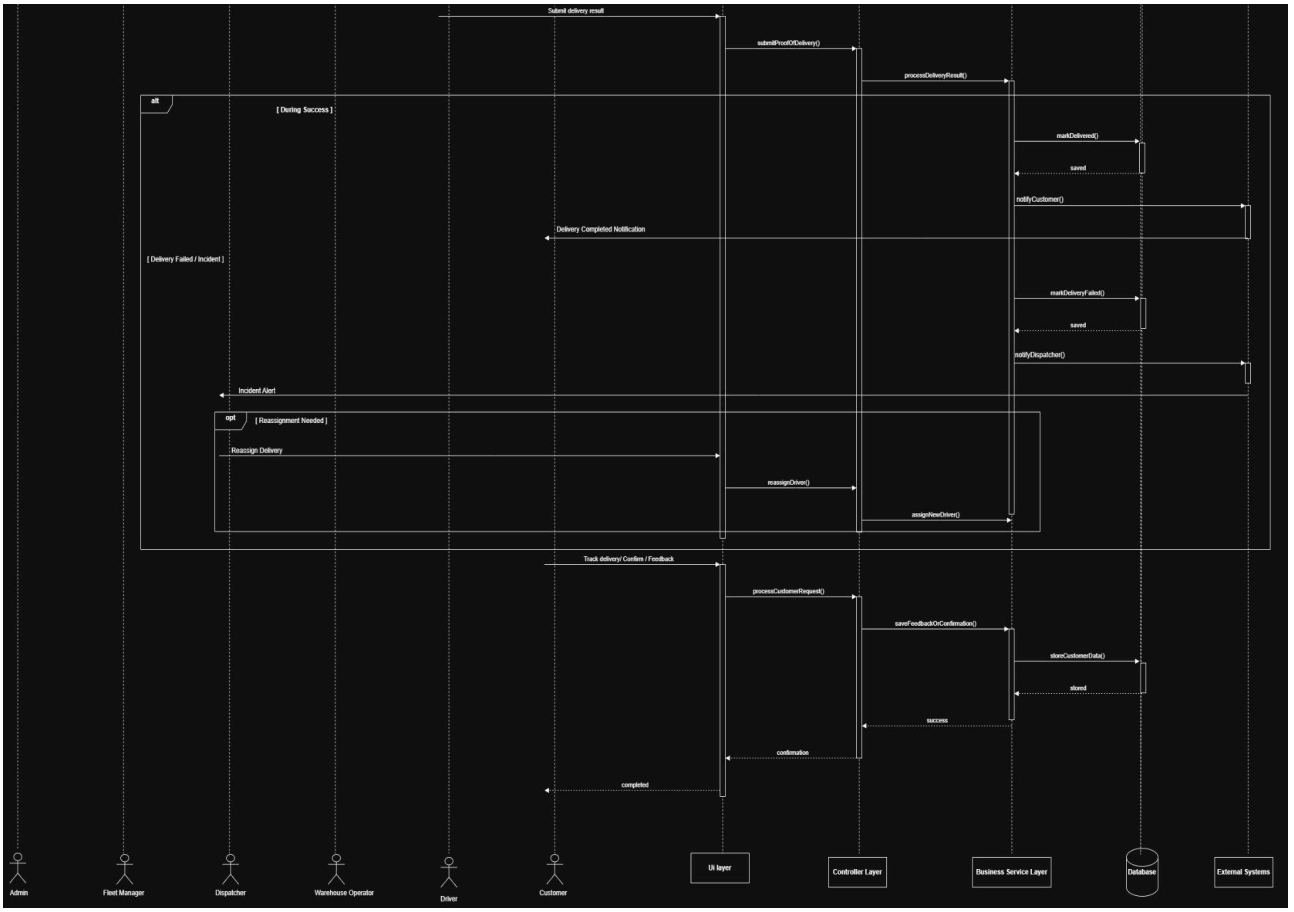


SEQUENCE DIAGRAM-






Check-in shift
startShift()
validateDriverCompliance()
checkDriverDocs()
driverValid
alt
[Valid Shift]
shiftApproved
[Invalid Shift]
shiftRejected
shiftStatus
Response
StartDelivery
startDelivery()
enableTracking()
startLiveTracking()
trackingStarted
loop
[During Delivery]
Send location update
updateLocation()
pushTrackingUpdate()
Live tracking Update


Submit delivery result
submitProofOfDelivery()
processDeliveryResult()
alt
[During Success]
markDelivered()
saved
notifyCustomer()
Delivery Completed Notification
[Delivery Failed / Incident]
markDeliveryFailed()
saved
notifyDispatcher()
Incident Alert
opt
[Reassignment Needed]
Reassign Delivery
reassignDriver()
assignNewDriver()
Track delivery/ Confirm / Feedback
processCustomerRequest()
saveFeedbackOrConfirmation()
storeCustomerData()
stored
success
confirmation
completed
Admin
Fleet Manager
Dispatcher
Warehouse Operator
Driver
Customer
UI layer
Controller Layer
Business Service Layer
Database
External Systems