

# Smart Logistics & Truck Management System

Project Specification Document

*Dammam Urban Development Challenge 2025*

**Reducing Port Traffic Congestion on King Abdulaziz Road  
Through AI-Powered Dynamic Permit Management**

*Challenge #2: Improving Mina Road Efficiency in Dammam*

**Team 1:** Admin Dashboard & Backend (Next.js 16)  
**Team 2:** Driver Mobile Application (React Native)  
**Team 3:** AI & Computer Vision (Python + YOLO)

February 2026

# Contents

<b>Abstract</b>	<b>4</b>
<b>1 Project Overview</b>	<b>5</b>
1.1 Problem Statement . . . . .	5
1.2 Proposed Solution . . . . .	5
1.3 Core Mechanism . . . . .	5
1.3.1 Priority Classification System . . . . .	6
1.4 System Architecture . . . . .	6
1.5 Key Innovations . . . . .	6
1.5.1 1. Priority-Aware Traffic Management . . . . .	7
1.5.2 2. Predictive Traffic Analysis . . . . .	7
1.5.3 3. Intelligent Rescheduling with Alternatives . . . . .	8
1.6 Port Integration Context . . . . .	8
1.6.1 Dammam Port Statistics . . . . .	8
1.6.2 Vessel Schedule Integration . . . . .	8
1.7 Alignment with Saudi Vision 2030 . . . . .	9
<b>2 Team Structure &amp; Responsibilities</b>	<b>10</b>
2.1 Team 1: Admin Dashboard & Core Backend . . . . .	10
2.1.1 Role: The Brain . . . . .	10
2.1.2 Tech Stack . . . . .	10
2.1.3 Core Responsibilities (Days 1-2) . . . . .	10
2.1.4 Enhanced Features (Days 3-4) . . . . .	11
2.1.5 Deliverables . . . . .	12
2.1.6 API Endpoints Summary . . . . .	12
2.2 Team 2: Driver Mobile Application . . . . .	13
2.2.1 Role: The Client . . . . .	13
2.2.2 Tech Stack . . . . .	13
2.2.3 Core Responsibilities (Days 1-2) . . . . .	13
2.2.4 Enhanced Features (Days 3-4) . . . . .	14
2.2.5 Screen Flow . . . . .	14
2.2.6 Deliverables . . . . .	15
2.3 Team 3: AI & Computer Vision . . . . .	16
2.3.1 Role: The Eye . . . . .	16
2.3.2 Recommended Tech Stack . . . . .	16
2.3.3 Core Responsibilities . . . . .	16
2.3.4 Technical Suggestions . . . . .	17
2.3.5 JSON Payload Contract . . . . .	18
2.3.6 Deliverables . . . . .	18
2.3.7 Decision Authority . . . . .	18

<b>3</b>	<b>Database Schema</b>	<b>19</b>
3.1	Core Tables	19
3.1.1	Drivers Table	19
3.1.2	Time Slots Table	19
3.1.3	Priority Rules Table	19
3.1.4	Permits Table	20
3.1.5	Vessel Schedules Table	21
3.1.6	Traffic Updates Table	21
3.1.7	Traffic Predictions Table	22
3.1.8	Driver Locations Table	22
3.1.9	Notifications Table	23
3.2	Database Functions	23
3.2.1	Auto-Update Timestamp	23
3.2.2	Slot Capacity Management	24
3.2.3	Priority-Aware Halting	24
3.3	Row Level Security	25
3.4	Seed Data	26
3.4.1	Time Slots with Predictions	26
3.4.2	Historical Traffic Data	26
3.5	Database Views	27
<b>4</b>	<b>Impact Analysis</b>	<b>29</b>
4.1	Overview	29
4.2	Priority Tier Framework	29
4.2.1	Classification System	29
4.2.2	Implementation Approach	29
4.3	Stakeholder Impact Analysis	30
4.3.1	Emergency Logistics (EMERGENCY Tier)	30
4.3.2	Commercial Time-Sensitive Operations (ESSENTIAL Tier)	30
4.3.3	Standard Freight Operations (NORMAL Tier)	31
4.3.4	Bulk Materials & Non-Urgent Freight (LOW Tier)	31
4.3.5	Port Operations & Maritime Logistics	31
4.3.6	Trucking Companies & Drivers	32
4.3.7	Commuters & Residents	33
4.4	Risk Analysis	33
4.4.1	System Abuse Risk	33
4.4.2	Commercial Liability Risk	34
4.4.3	Technical Risks	34
4.5	Cost-Benefit Analysis	35
4.5.1	Implementation Costs	35
4.5.2	Economic Benefits	35
4.6	Social & Environmental Impact	36
4.6.1	Quality of Life Improvements	36
4.6.2	Environmental Benefits	36
4.7	Conclusion	37

<b>5</b>	<b>Mitigation Strategies</b>	<b>38</b>
5.1	Priority Tier System . . . . .	38
5.2	Predictive Warnings . . . . .	38
5.3	Compensation . . . . .	38
5.4	Alternative Routes . . . . .	38
5.5	Priority System Abuse Mitigation . . . . .	39
5.5.1	Challenge . . . . .	39
5.5.2	Why This Matters . . . . .	39
5.5.3	Mitigation Strategies . . . . .	39
5.5.4	Expected Outcomes . . . . .	40
5.5.5	Contingency Plan . . . . .	40
5.6	Gradual Rollout . . . . .	40
5.7	Stakeholder Engagement . . . . .	41
<b>6</b>	<b>Success Metrics</b>	<b>42</b>
6.1	Traffic Metrics . . . . .	42
6.2	Driver Metrics . . . . .	42
6.3	System Performance . . . . .	42
6.4	Enhanced Features Metrics . . . . .	42
6.5	Port Operations Metrics . . . . .	42
<b>7</b>	<b>Future Work &amp; Enhancement Roadmap</b>	<b>43</b>
7.1	Prototype vs Production Context . . . . .	43
7.1.1	Current Hackathon Prototype Scope . . . . .	43
7.1.2	Prototype Limitations . . . . .	44
7.1.3	Production Deployment Requirements . . . . .	44
7.2	Short-Term (6-12 months) . . . . .	44
7.3	Medium-Term (1-2 years) . . . . .	45
7.4	Long-Term (2+ years) . . . . .	45
<b>8</b>	<b>Conclusion</b>	<b>46</b>
8.1	Executive Summary . . . . .	46
8.2	Key Takeaways . . . . .	46
8.3	Success Factors . . . . .	46
8.4	Hackathon Prototype vs Production System . . . . .	47
8.4.1	Current Prototype Scope . . . . .	47
8.4.2	Production Deployment Requirements . . . . .	47
8.4.3	Phased Implementation Timeline . . . . .	48
8.5	Final Recommendation . . . . .	48

## Abstract

This document outlines the comprehensive specification for a Smart Logistics & Truck Management System designed to reduce traffic congestion on King Abdulaziz Road (Mina Road) in Dammam, the primary corridor for port-bound cargo traffic. Developed as part of the Dammam Urban Development Hackathon 2025 (Challenge #2), the system addresses the unique challenges of managing approximately 1,260 daily truck movements to and from Dammam seaport.

The system employs real-time computer vision analysis, priority-aware dynamic permit management, predictive traffic analytics, and mobile notifications to optimize truck entry/exit times based on current traffic conditions and cargo urgency. Unlike traditional permit systems that treat all trucks equally, this solution implements a four-tier priority classification (EMERGENCY, ESSENTIAL, NORMAL, LOW) ensuring critical cargo maintains flow during congestion while managing overall traffic load.

Key innovations include intelligent rescheduling with alternative time slot suggestions, historical pattern-based traffic prediction, and vessel schedule coordination. The system protects time-sensitive operations such as medical supplies and perishable goods while balancing port efficiency with commuter traffic flow.

The project is structured as a polyrepository system with three distinct teams: Backend/Dashboard (Next.js), Mobile Application (React Native), and AI/Computer Vision (Python). This specification covers team responsibilities, technical architecture, enhanced database schema, API endpoints, port integration requirements, and critical analysis of impacts on maritime logistics operations.

**Keywords:** Port Logistics, Traffic Management, Computer Vision, YOLO, Smart Cities, Maritime Cargo, Priority Management, Predictive Analytics, Dammam Port, Real-time Systems, Vision 2030

# 1 Project Overview

## 1.1 Problem Statement

King Abdulaziz Road (Port Road/Mina Road) in Dammam serves as the primary corridor for truck traffic to and from Dammam seaport. With approximately **1,260 trucks daily** (4,310 trucks entering port and 4,538 trucks exiting port per week), the road experiences severe traffic congestion during peak hours, particularly affecting both port-bound cargo trucks and regular commuter traffic.

This challenge was identified as part of the **Dammam Urban Development Hackathon 2025 (Challenge #2)**, recognizing the critical need to balance efficient port logistics with urban mobility and resident quality of life.

The current truck permit system operates on a static, pre-scheduled basis where once a truck receives a permit for a specific time (e.g., 8:00 AM), it enters the road regardless of real-time traffic conditions or cargo urgency.

### Current System Issues

- Trucks entering during rush hour worsen congestion
- All trucks treated equally regardless of cargo urgency
- No coordination with ship arrival schedules
- Unpredictable delays for both trucks and regular vehicles
- Increased accident rates due to congestion
- Economic losses from fuel waste and missed delivery schedules
- Environmental impact from increased emissions and idling
- Port demurrage costs when cargo delays occur

## 1.2 Proposed Solution

A dynamic, AI-powered permit management system with intelligent priority handling that:

1. **Monitors traffic in real-time** using computer vision cameras
2. **Analyzes vehicle density** distinguishing between cars and trucks
3. **Classifies traffic status** (NORMAL, MODERATE, CONGESTED)
4. **Prioritizes permits** based on cargo type and urgency
5. **Predicts congestion** using historical patterns and vessel schedules
6. **Automatically adjusts permits** based on current conditions
7. **Suggests alternative time slots** when permits are rescheduled
8. **Notifies drivers instantly** via mobile application
9. **Tracks compliance** through GPS location monitoring
10. **Provides analytics** for traffic pattern optimization

## 1.3 Core Mechanism

The system operates on a traffic-responsive ticketing mechanism with priority-aware management:

- **NORMAL Traffic:** Approve all new permit requests across all priority levels
- **MODERATE Traffic:** Issue warnings, prioritize EMERGENCY and ESSENTIAL cargo
- **CONGESTED Traffic:** Halt NORMAL and LOW priority permits only, maintain flow for urgent cargo

### 1.3.1 Priority Classification System

Unlike traditional systems that treat all trucks equally, this system implements a four-tier priority classification that protects time-sensitive operations while managing overall traffic load:

Priority	Cargo Type	Max Delay	Congestion Response
EMERGENCY	Perishable goods (food, medicine), medical supplies, vaccines	0 minutes	Never halted
ESSENTIAL	Time-sensitive cargo, JIT manufacturing parts, scheduled deliveries	2 hours	Rarely delayed
NORMAL	Standard containers, general cargo, regular freight	8 hours	Can be delayed
LOW	Bulk materials, non-urgent freight, storage-bound goods	24 hours	First to reschedule
sm			

Table 1: Priority Tier System

#### Example Scenarios:

- **Emergency:** Truck carrying vaccines for hospital → Approved even during severe congestion
- **Essential:** Automotive parts for JIT manufacturing → Delayed maximum 2 hours
- **Normal:** Standard shipping containers → Can wait until traffic clears
- **Low:** Bulk cement for warehouse storage → Rescheduled to off-peak hours

## 1.4 System Architecture

## 1.5 Key Innovations

This system introduces three key innovations that distinguish it from traditional traffic management approaches:

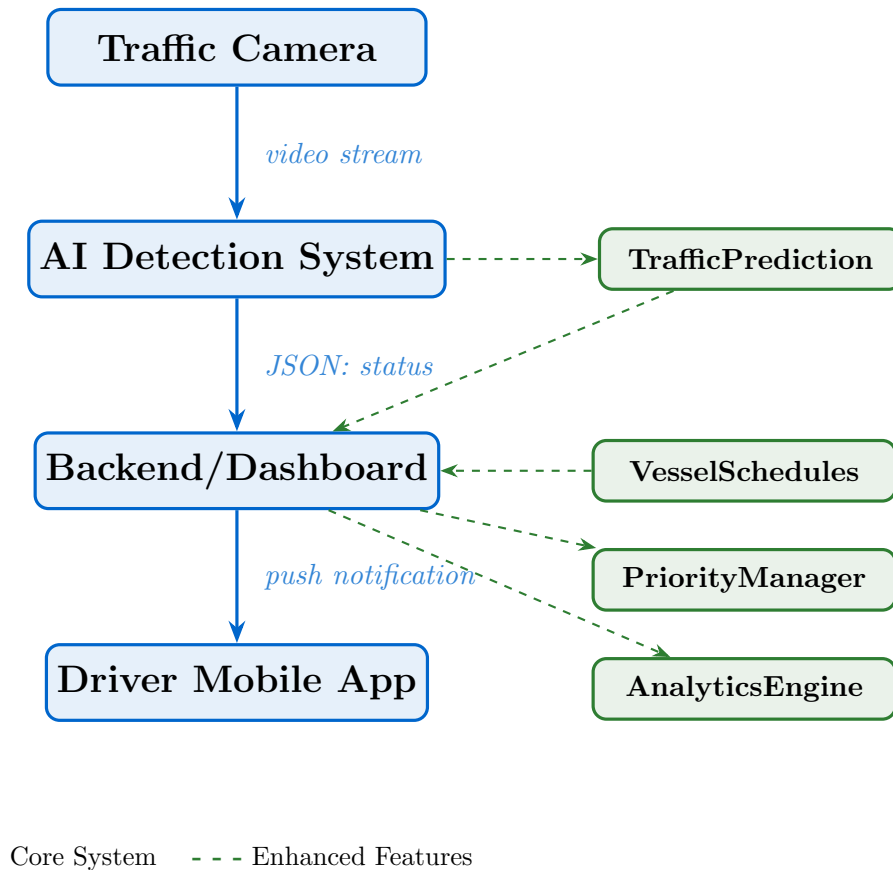


Figure 1: Enhanced System Architecture

### 1.5.1 1. Priority-Aware Traffic Management

Traditional systems use "first come, first served" logic that treats all trucks identically. This creates situations where a truck carrying life-saving medical supplies can be blocked while bulk materials pass through.

**Our approach:** Cargo urgency determines traffic priority. The system continuously evaluates: *"Do we have capacity for more trucks right now, and what type of cargo is most critical?"*

**Real-world impact:** A refrigerated truck carrying vaccines never faces delays, while a truck carrying gravel for a construction site can wait until off-peak hours.

### 1.5.2 2. Predictive Traffic Analysis

Instead of only reacting to current congestion, the system predicts traffic patterns based on:

- Historical traffic data (7-day patterns)
- Vessel arrival schedules (expected truck surges)
- Time-of-day factors (rush hours, prayer times)
- Day-of-week patterns (weekends vs weekdays)

**Example prediction:**

Tuesday 3:00 PM prediction:

- Current status: NORMAL (85 vehicles)
- Predicted 5:00 PM: CONGESTED (145 vehicles)
- Reason: Evening rush hour + 2 vessels unloading
- Recommendation: Reduce permit approvals starting 4:30 PM

### 1.5.3 3. Intelligent Rescheduling with Alternatives

When traffic conditions force permit delays, drivers don't receive a generic "permit denied" message. Instead, the system:

1. Analyzes the driver's original time slot
2. Finds 2-3 alternative slots within 4 hours
3. Checks predicted traffic status for each alternative
4. Presents options ranked by convenience and traffic likelihood
5. Allows one-tap rebooking

#### Example notification:

"Heavy traffic detected. Your 3:00 PM slot has been delayed.

Alternative times available: - 4:30 PM - 6:30 PM (Light traffic predicted) -  
6:30 PM - 8:30 PM (Moderate traffic possible) - Tomorrow 10:00 AM - 12:00  
PM (Clear roads expected)

Tap to select new time."

## 1.6 Port Integration Context

### 1.6.1 Dammam Port Statistics

Based on current port operations data from the Dammam Urban Development Hackathon 2025:

Metric	Volume
Trucks entering port weekly	4,310 trucks
Trucks exiting port weekly	4,538 trucks
Total weekly truck movements	~8,848 trucks
Average daily movements	~1,260 trucks
Peak hour concentration	40-50% of daily volume
Typical peak hours	7-9 AM, 4-6 PM

Table 2: Current Truck Traffic Statistics - Dammam Port

### 1.6.2 Vessel Schedule Integration

The system includes a vessel schedule display that shows upcoming ship arrivals and expected truck volumes. While the hackathon prototype uses simulated data, production deployment would integrate with Saudi Ports Authority (Mawani) systems.

#### Prototype approach:

- Manual entry of vessel schedules for demonstration
- Simulated arrival times and truck estimates
- Dashboard visualization of upcoming traffic surges

**Production vision (future work):**

- Real-time Mawani API integration
- Automatic truck count estimation from cargo manifests
- Coordination with customs clearance systems
- Integration with port operational schedules

## 1.7 Alignment with Saudi Vision 2030

This project directly supports multiple Vision 2030 objectives:

### 1. Logistics Hub Development

- Improve port efficiency and throughput
- Position Saudi Arabia as regional logistics leader
- Support Kingdom's trade facilitation goals

### 2. Digital Transformation

- AI/ML application in public infrastructure
- Smart city initiatives in Dammam
- Government-industry data integration

### 3. Quality of Life

- Reduce urban congestion for residents
- Improve air quality in residential areas
- Enhance citizen services through technology

### 4. Environmental Sustainability

- 15-20% reduction in vehicle emissions
- Fuel efficiency improvements
- Green logistics practices

## 2 Team Structure & Responsibilities

The project utilizes a **polyrepository architecture** with three independent teams, each maintaining their own Git repository.

### 2.1 Team 1: Admin Dashboard & Core Backend

#### 2.1.1 Role: The Brain

Team 1 serves as the central intelligence of the system, managing all business logic, data persistence, administrative interfaces, and enhanced features including predictive analytics and priority management.

#### 2.1.2 Tech Stack

Component	Technology
Framework	Next.js 16 (App Router)
Backend	Server Actions + REST API
Database	Supabase (PostgreSQL)
Authentication	Supabase Auth (Phone OTP)
Real-time	Supabase Realtime Subscriptions
State Management	Zustand
Validation	Zod
Styling	Tailwind CSS
Charts	Recharts
Deployment	Vercel

#### 2.1.3 Core Responsibilities (Days 1-2)

##### 1. Backend API Development

- RESTful endpoints for all system operations
- Receive and process traffic updates from AI team
- Permit booking and management logic
- Push notification dispatch

##### 2. Database Management

- Schema design and implementation
- Data integrity and relationships
- Real-time subscriptions configuration
- Row-level security policies

##### 3. Admin Dashboard - Basic

- Real-time traffic monitoring interface
- Basic permit management
- Traffic status display

- Simple statistics

#### 4. Business Logic

- Automatic permit status updates based on traffic
- Basic notification triggering
- Time slot capacity management

### 2.1.4 Enhanced Features (Days 3-4)

#### 1. Priority-Based Permit Management

- Implement four-tier priority system (EMERGENCY, ESSENTIAL, NORMAL, LOW)
- Priority-aware congestion response logic
- Cargo type classification
- Smart halting that protects urgent cargo

#### 2. Vessel Schedule Integration

- Vessel schedule database management
- Dashboard widget showing upcoming arrivals
- Expected truck volume display
- Manual vessel entry interface for demo

#### 3. Traffic Prediction System

- Heuristic-based prediction algorithm
- Historical pattern analysis (7-day data)
- Vessel arrival impact calculation
- Confidence scoring
- Prediction display in dashboard

#### 4. Analytics Dashboard

- Daily permit statistics
- Traffic pattern charts (Recharts)
- Average delay calculations
- Emergency cargo success rate tracking
- Historical trend visualization

#### 5. Rescheduling Suggestions API

- Alternative slot finding algorithm
- Traffic prediction for alternative times
- Ranking by convenience and availability
- One-tap rebooking endpoint

#### 6. Enhanced Map Features

- Interactive map with Leaflet.js

- Real-time driver location markers
- Camera location markers
- ETA display for incoming trucks
- Color-coded priority indicators

### 2.1.5 Deliverables

#### Team 1 Must Deliver

1. Deployed backend API with all endpoints operational
2. Admin dashboard with real-time updates
3. Complete database schema with sample data
4. **Priority tier management system**
5. **Vessel schedule widget showing upcoming arrivals**
6. **Traffic prediction display with confidence scores**
7. **Analytics dashboard with charts and metrics**
8. **Rescheduling suggestions API**
9. Interactive map showing driver locations
10. API documentation (endpoints, request/response formats)
11. Integration testing suite

### 2.1.6 API Endpoints Summary

#### Core Endpoints:

- POST /api/traffic - Receive AI updates
- GET /api/slots - Get available time slots
- POST /api/book - Book a permit
- GET /api/permits - Get driver permits

#### Enhanced Endpoints:

- GET /api/vessels - Get vessel schedule
- GET /api/predict - Get traffic predictions
- GET /api/slots/alternatives - Get alternative time slots
- GET /api/analytics/daily - Get daily statistics
- POST /api/permits/reschedule - Reschedule a permit

## 2.2 Team 2: Driver Mobile Application

### 2.2.1 Role: The Client

Team 2 develops the end-user interface for truck drivers to manage their permits, receive notifications, track their compliance status, and handle rescheduling when traffic conditions change.

### 2.2.2 Tech Stack

Component	Technology
Framework	React Native (Expo)
Navigation	Expo Router
Backend Client	Supabase JavaScript Client
Authentication	Supabase Auth
Location Services	expo-location
Notifications	expo-notifications
QR Code	@kichiyaki/react-native-barcode-generator
Styling	NativeWind (Tailwind for RN)
State Management	Zustand

### 2.2.3 Core Responsibilities (Days 1-2)

#### 1. Authentication System

- Phone number login with OTP verification
- User profile creation (name, vehicle plate)
- Session management

#### 2. Permit Booking Interface

- Date picker for slot selection
- Real-time availability display
- Booking confirmation flow
- Multiple permit management

#### 3. Permit Display & QR Codes

- Generate unique QR codes per permit
- Display permit status (APPROVED, HALTED, etc.)
- Show time slot details
- Status change animations

#### 4. GPS Tracking

- Request location permissions
- Periodic location updates (every 30 seconds)
- Visual indicator of tracking status
- Background location support

## 5. Push Notifications

- Register device for notifications
- Handle incoming notification display
- Navigate to relevant screens on tap
- Notification history

### 2.2.4 Enhanced Features (Days 3-4)

#### 1. Priority Tier Selection

- Cargo type dropdown during booking
- Priority tier explanation screen
- Visual priority badges (color-coded)
- Priority tier display on permit card

#### 2. Alternative Slots Screen

- Display when permit is halted
- Show 2-3 alternative time slots
- Traffic prediction for each alternative
- One-tap rebooking functionality
- Visual comparison of alternatives

#### 3. Enhanced Notifications

- Priority-specific notification styling
- Inline action buttons (View Alternatives, Confirm)
- Rich notification content
- Notification grouping by permit

#### 4. ETA Calculation (Optional)

- Calculate distance to port
- Estimate arrival time
- Send ETA to backend
- Display ETA on permit card

### 2.2.5 Screen Flow

#### Core Screens:

1. Login / OTP Verification
2. Profile Setup
3. Home (Active Permits)
4. Book Permit (Date + Time + Priority)
5. Permit Details (QR + Status)
6. Settings

#### Enhanced Screens:

1. Priority Selection Modal
2. Alternative Slots Screen
3. Permit Halted Notification
4. Rescheduling Confirmation

### 2.2.6 Deliverables

#### Team 2 Must Deliver

1. Functional mobile application (iOS & Android)
2. Complete authentication flow
3. Permit booking and display system
4. **Priority tier selection during booking**
5. **Alternative slots screen when permit halted**
6. **One-tap rescheduling functionality**
7. GPS tracking implementation
8. Push notification integration
9. QR code generation and display
10. APK/IPA for demo purposes
11. Screen recording of all features (3-4 minutes)

## 2.3 Team 3: AI & Computer Vision

### 2.3.1 Role: The Eye

Team 3 develops the intelligence layer that analyzes traffic conditions through computer vision, providing real-time data to inform permit decisions. **Team 3's responsibilities remain unchanged** from the core system specification.

### 2.3.2 Recommended Tech Stack

#### Note: Flexible Technical Decisions

The following represents recommended technologies. Team 3 has full autonomy to select alternative approaches that better suit their expertise or project requirements.

Component	Suggested Technology
Programming Language	Python 3.10+
Object Detection	Ultralytics YOLOv8 or YOLOv11
Video Processing	OpenCV
HTTP Client	requests library
Visualization	matplotlib, cv2.imshow
Model Framework	PyTorch (via Ultralytics)
Deployment	FastAPI, Flask, or script

### 2.3.3 Core Responsibilities

#### 1. Video Input Processing

- Accept video files or streams as input
- Handle various video formats and resolutions
- Frame extraction and preprocessing
- Performance optimization for real-time processing

#### 2. Vehicle Detection & Classification

- Detect vehicles in each frame
- Classify vehicles (cars vs. trucks/buses)
- Count total vehicles and specific truck count
- Handle occlusions and varying lighting conditions

#### 3. Traffic Status Classification

- Define traffic density thresholds
- Calculate traffic status (NORMAL/MODERATE/CONGESTED)
- Generate recommendations
- Adjust thresholds based on data (optional)

#### 4. Backend Integration

- Send JSON payloads to Team 1's API
- Implement retry logic for failed requests
- Maintain consistent update intervals
- Log all detections and API responses

## 5. Demo Visualization

- Create annotated videos with bounding boxes
- Overlay vehicle counts on frames
- Display current traffic status
- Generate comparison videos

### 2.3.4 Technical Suggestions

#### Team 3 Autonomy

These are recommendations. Team 3 should evaluate and choose approaches that align with their expertise.

#### Model Selection:

- **YOLOv8n:** Fastest, 30+ FPS
- **YOLOv8s/m:** Balanced speed and accuracy
- **YOLOv11:** Latest with improved accuracy
- **Custom:** Train or fine-tune if needed

#### Transfer Learning: COCO dataset pre-trained classes:

- Class 2: car
- Class 3: motorcycle
- Class 5: bus
- Class 7: truck

```
1 def classify_traffic(vehicle_count, truck_count):
2     if vehicle_count > 150 or truck_count > 15:
3         return "CONGESTED", "HALT_TRUCK_PERMITS"
4     elif vehicle_count > 100 or truck_count > 12:
5         return "MODERATE", "LIMIT_PERMITS"
6     else:
7         return "NORMAL", "ALLOW_PERMITS"
```

Listing 1: Sample Classification

### 2.3.5 JSON Payload Contract

Traffic Classification Example: Required Format (Mandatory):

```
1 {  
2   "camera_id": "CAM_01_KING_ABDULAZIZ",  
3   "timestamp": "2026-02-05T20:30:00Z",  
4   "status": "CONGESTED",  
5   "vehicle_count": 145,  
6   "truck_count": 12,  
7   "recommendation": "HALT_TRUCK_PERMITS"  
8 }
```

Listing 2: API Payload

### 2.3.6 Deliverables

#### Team 3 Must Deliver

1. Working vehicle detection system
2. Traffic classification algorithm
3. Integration with Team 1's API
4. 3 demo videos (NORMAL, MODERATE, CONGESTED)
5. Annotated videos with bounding boxes
6. Documentation of thresholds

### 2.3.7 Decision Authority

#### Team 3 Decisions

Team 3 decides:

- Which YOLO version
- Transfer learning vs custom training
- Frame processing strategy
- Traffic thresholds
- Video sources
- Optimization techniques

**Only requirement:** JSON format to Team 1's API

## 3 Database Schema

### 3.1 Core Tables

#### 3.1.1 Drivers Table

```
1 CREATE TABLE drivers (  
2   id uuid PRIMARY KEY  
3   REFERENCES auth.users(id) ON DELETE CASCADE,  
4   phone text UNIQUE NOT NULL,  
5   name text NOT NULL,  
6   vehicle_plate text NOT NULL,  
7   vehicle_type text DEFAULT 'TRUCK',  
8   push_token text,  
9   is_active boolean DEFAULT true,  
10  created_at timestamptz DEFAULT now(),  
11  updated_at timestamptz DEFAULT now()  
12 );  
13  
14 CREATE INDEX idx_drivers_phone ON drivers(phone);
```

Listing 3: Drivers Schema

#### 3.1.2 Time Slots Table

```
1 CREATE TABLE time_slots (  
2   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),  
3   date date NOT NULL,  
4   start_time time NOT NULL,  
5   end_time time NOT NULL,  
6   capacity int DEFAULT 10,  
7   booked int DEFAULT 0,  
8   status text DEFAULT 'AVAILABLE',  
9   predicted_traffic text, -- NORMAL, MODERATE, CONGESTED  
10  created_at timestamptz DEFAULT now(),  
11  UNIQUE(date, start_time)  
12 );  
13  
14 CREATE INDEX idx_time_slots_date ON time_slots(date);
```

Listing 4: Time Slots Schema

#### 3.1.3 Priority Rules Table

```
1 CREATE TABLE priority_rules (  
2   id serial PRIMARY KEY,  
3   cargo_type text UNIQUE NOT NULL,  
4   priority_level text NOT NULL,  
5   max_delay_minutes int DEFAULT 0,  
6   can_be_halted boolean DEFAULT true,  
7   description text,  
8   color_code text -- For UI display  
9 );
```

```

10
11 -- Seed priority rules
12 INSERT INTO priority_rules
13   (cargo_type, priority_level, max_delay_minutes, can_be_halted,
14    description, color_code)
15 VALUES
16   ('PERISHABLE', 'EMERGENCY', 0, false,
17    'Food, medicine - expires within hours', '#EF4444'),
18   ('MEDICAL', 'EMERGENCY', 0, false,
19    'Medical supplies, vaccines, blood products', '#EF4444'),
20   ('TIME_SENSITIVE', 'ESSENTIAL', 120, false,
21    'JIT manufacturing, scheduled deliveries', '#F59E0B'),
22   ('STANDARD', 'NORMAL', 480, true,
23    'Regular containers, general cargo', '#3B82F6'),
24   ('BULK', 'LOW', 1440, true,
25    'Bulk materials, non-urgent freight', '#6B7280');
26
27 CREATE INDEX idx_priority_cargo_type
28   ON priority_rules(cargo_type);

```

Listing 5: Priority Rules Schema

### 3.1.4 Permits Table

```

1 CREATE TABLE permits (
2   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3   driver_id uuid REFERENCES drivers(id) ON DELETE CASCADE,
4   slot_id uuid REFERENCES time_slots(id) ON DELETE CASCADE,
5   vessel_id uuid REFERENCES vessel_schedules(id),
6   qr_code text UNIQUE NOT NULL,
7   status text DEFAULT 'APPROVED',
8   priority text DEFAULT 'NORMAL',
9   cargo_type text DEFAULT 'STANDARD',
10  original_slot_id uuid REFERENCES time_slots(id),
11  rescheduled_count int DEFAULT 0,
12  suggested_slots jsonb,
13  created_at timestamptz DEFAULT now(),
14  updated_at timestamptz DEFAULT now(),
15  expires_at timestamptz,
16  entry_time timestamptz,
17  exit_time timestamptz,
18  notes text
19 );
20
21 CREATE INDEX idx_permits_driver ON permits(driver_id);
22 CREATE INDEX idx_permits_status ON permits(status);
23 CREATE INDEX idx_permits_priority ON permits(priority);
24 CREATE INDEX idx_permits_cargo ON permits(cargo_type);
25 CREATE INDEX idx_permits_vessel ON permits(vessel_id);

```

Listing 6: Enhanced Permits Schema

**Status Values:** PENDING, APPROVED, HALTED, CANCELLED, EXPIRED, COMPLETED

**Priority Values:** EMERGENCY, ESSENTIAL, NORMAL, LOW

**Cargo Type Values:** PERISHABLE, MEDICAL, TIME\_SENSITIVE, STANDARD, BULK

### 3.1.5 Vessel Schedules Table

```

1 CREATE TABLE vessel_schedules (
2   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3   vessel_name text NOT NULL,
4   arrival_date date NOT NULL,
5   arrival_time time,
6   estimated_trucks int DEFAULT 0,
7   actual_trucks int DEFAULT 0,
8   status text DEFAULT 'SCHEDULED',
9   cargo_priority text DEFAULT 'NORMAL',
10  notes text,
11  created_at timestamptz DEFAULT now(),
12  updated_at timestamptz DEFAULT now()
13 );
14
15 CREATE INDEX idx_vessel_arrival
16   ON vessel_schedules(arrival_date);
17 CREATE INDEX idx_vessel_status
18   ON vessel_schedules(status);
19
20 -- Demo data for presentation
21 INSERT INTO vessel_schedules
22   (vessel_name, arrival_date, arrival_time,
23    estimated_trucks, status, cargo_priority)
24 VALUES
25   ('MSC TOKYO', CURRENT_DATE + 1, '08:00',
26    180, 'SCHEDULED', 'NORMAL'),
27   ('MAERSK SHANGHAI', CURRENT_DATE + 1, '14:00',
28    220, 'SCHEDULED', 'NORMAL'),
29   ('COSCO PHOENIX', CURRENT_DATE + 2, '06:00',
30    150, 'SCHEDULED', 'TIME_SENSITIVE'),
31   ('EVERGREEN HARMONY', CURRENT_DATE + 2, '15:00',
32    190, 'SCHEDULED', 'NORMAL'),
33   ('CMA CGM PEARL', CURRENT_DATE + 3, '10:00',
34    160, 'SCHEDULED', 'PERISHABLE');

```

Listing 7: Vessel Schedules Schema

### 3.1.6 Traffic Updates Table

```

1 CREATE TABLE traffic_updates (
2   id serial PRIMARY KEY,
3   camera_id text NOT NULL,
4   timestamp timestamptz DEFAULT now(),
5   vehicle_count int NOT NULL,
6   truck_count int NOT NULL,
7   car_count int,
8   status text NOT NULL,
9   density_score decimal(5,2),
10  recommendation text,

```

```

11     created_at timestamptz DEFAULT now()
12 );
13
14 CREATE INDEX idx_traffic_timestamp
15     ON traffic_updates(timestamp DESC);
16 CREATE INDEX idx_traffic_camera
17     ON traffic_updates(camera_id);
18 CREATE INDEX idx_traffic_status
19     ON traffic_updates(status);

```

Listing 8: Traffic Updates Schema

### 3.1.7 Traffic Predictions Table

```

1 CREATE TABLE traffic_predictions (
2     id serial PRIMARY KEY,
3     camera_id text NOT NULL,
4     prediction_time timestamptz NOT NULL,
5     predicted_for timestamptz NOT NULL,
6     predicted_status text NOT NULL,
7     confidence_score decimal(5,2),
8     factors jsonb,
9     created_at timestamptz DEFAULT now()
10 );
11
12 CREATE INDEX idx_predictions_time
13     ON traffic_predictions(prediction_time DESC);
14 CREATE INDEX idx_predictions_for
15     ON traffic_predictions(predicted_for);
16
17 -- Example factors JSON structure:
18 -- {
19 --     "rush_hour": true,
20 --     "vessel_arrivals": 2,
21 --     "weekend": false,
22 --     "historical_pattern": "CONGESTED"
23 -- }

```

Listing 9: Traffic Predictions Schema

### 3.1.8 Driver Locations Table

```

1 CREATE TABLE driver_locations (
2     id serial PRIMARY KEY,
3     driver_id uuid REFERENCES drivers(id) ON DELETE CASCADE,
4     permit_id uuid REFERENCES permits(id) ON DELETE CASCADE,
5     latitude decimal(10, 8) NOT NULL,
6     longitude decimal(11, 8) NOT NULL,
7     accuracy decimal(10, 2),
8     speed decimal(5, 2),
9     eta_minutes int, -- Estimated time to arrival
10    recorded_at timestamptz DEFAULT now()
11 );
12

```

```

13 CREATE INDEX idx_locations_driver
14     ON driver_locations(driver_id);
15 CREATE INDEX idx_locations_time
16     ON driver_locations(recorded_at DESC);
17 CREATE INDEX idx_locations_permit
18     ON driver_locations(permit_id);

```

Listing 10: GPS Tracking Schema

### 3.1.9 Notifications Table

```

1 CREATE TABLE notifications (
2     id serial PRIMARY KEY,
3     driver_id uuid REFERENCES drivers(id) ON DELETE CASCADE,
4     permit_id uuid REFERENCES permits(id) ON DELETE SET NULL,
5     title text NOT NULL,
6     body text NOT NULL,
7     type text DEFAULT 'INFO',
8     data jsonb,
9     sent_at timestamptz DEFAULT now(),
10    status text DEFAULT 'SENT',
11    error_message text
12 );
13
14 CREATE INDEX idx_notifications_driver
15     ON notifications(driver_id);
16 CREATE INDEX idx_notifications_type
17     ON notifications(type);
18
19 -- Notification types:
20 -- INFO, WARNING, URGENT, RESCHEDULE, APPROVAL, DENIAL

```

Listing 11: Notifications Log Schema

## 3.2 Database Functions

### 3.2.1 Auto-Update Timestamp

```

1 CREATE OR REPLACE FUNCTION update_updated_at_column()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     NEW.updated_at = now();
5     RETURN NEW;
6 END;
7 $$ LANGUAGE plpgsql;
8
9 CREATE TRIGGER update_drivers_updated_at
10 BEFORE UPDATE ON drivers
11 FOR EACH ROW
12 EXECUTE FUNCTION update_updated_at_column();
13
14 CREATE TRIGGER update_permits_updated_at
15 BEFORE UPDATE ON permits
16 FOR EACH ROW

```

```

17 EXECUTE FUNCTION update_updated_at_column();
18
19 CREATE TRIGGER update_vessels_updated_at
20 BEFORE UPDATE ON vessel_schedules
21 FOR EACH ROW
22 EXECUTE FUNCTION update_updated_at_column();

```

Listing 12: Timestamp Trigger

### 3.2.2 Slot Capacity Management

```

1 CREATE OR REPLACE FUNCTION update_slot_booking_count()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     IF TG_OP = 'INSERT' AND NEW.status = 'APPROVED' THEN
5         UPDATE time_slots
6         SET booked = booked + 1,
7             status = CASE
8                 WHEN booked + 1 >= capacity
9                 THEN 'FULL' ELSE 'AVAILABLE' END
10        WHERE id = NEW.slot_id;
11    ELSIF TG_OP = 'UPDATE'
12    AND OLD.status = 'APPROVED'
13    AND NEW.status != 'APPROVED' THEN
14        UPDATE time_slots
15        SET booked = booked - 1,
16            status = CASE
17                WHEN booked - 1 < capacity
18                THEN 'AVAILABLE' ELSE status END
19        WHERE id = NEW.slot_id;
20    END IF;
21    RETURN NEW;
22 END;
23 $$ LANGUAGE plpgsql;
24
25 CREATE TRIGGER update_slot_count
26 AFTER INSERT OR UPDATE ON permits
27 FOR EACH ROW
28 EXECUTE FUNCTION update_slot_booking_count();

```

Listing 13: Booking Count Trigger

### 3.2.3 Priority-Aware Halting

```

1 CREATE OR REPLACE FUNCTION halt_permits_by_priority(
2     traffic_status text
3 )
4 RETURNS TABLE(halted_count int, protected_count int) AS $$
5 DECLARE
6     halted int := 0;
7     protected int := 0;
8 BEGIN
9     -- Only halt NORMAL and LOW priority permits
10    IF traffic_status = 'CONGESTED' THEN

```

```

11 UPDATE permits
12 SET status = 'HALTED',
13     updated_at = now()
14 WHERE status = 'APPROVED'
15     AND priority IN ('NORMAL', 'LOW')
16     AND expires_at > now();
17
18 GET DIAGNOSTICS halted = ROW_COUNT;
19
20 -- Count protected permits
21 SELECT COUNT(*) INTO protected
22 FROM permits
23 WHERE status = 'APPROVED'
24     AND priority IN ('EMERGENCY', 'ESSENTIAL')
25     AND expires_at > now();
26 END IF;
27
28 RETURN QUERY SELECT halted, protected;
29 END;
30 $$ LANGUAGE plpgsql;

```

Listing 14: Smart Halting Function

### 3.3 Row Level Security

```

1 ALTER TABLE drivers ENABLE ROW LEVEL SECURITY;
2 ALTER TABLE permits ENABLE ROW LEVEL SECURITY;
3 ALTER TABLE driver_locations ENABLE ROW LEVEL SECURITY;
4
5 -- Drivers view own data
6 CREATE POLICY "Drivers can view own profile" ON drivers
7     FOR SELECT USING (auth.uid() = id);
8
9 CREATE POLICY "Drivers can update own profile" ON drivers
10    FOR UPDATE USING (auth.uid() = id);
11
12 -- Drivers view own permits
13 CREATE POLICY "Drivers can view own permits" ON permits
14    FOR SELECT USING (auth.uid() = driver_id);
15
16 -- Drivers insert own locations
17 CREATE POLICY "Drivers insert own locations"
18    ON driver_locations
19    FOR INSERT WITH CHECK (auth.uid() = driver_id);
20
21 -- Public reads
22 CREATE POLICY "Public view time slots" ON time_slots
23    FOR SELECT USING (true);
24
25 CREATE POLICY "Public view traffic" ON traffic_updates
26    FOR SELECT USING (true);
27
28 CREATE POLICY "Public view vessels" ON vessel_schedules
29    FOR SELECT USING (true);
30

```

```

31 CREATE POLICY "Public view priority rules" ON priority_rules
32   FOR SELECT USING (true);
33
34 CREATE POLICY "Public view predictions" ON traffic_predictions
35   FOR SELECT USING (true);

```

Listing 15: RLS Policies

## 3.4 Seed Data

### 3.4.1 Time Slots with Predictions

```

1 INSERT INTO time_slots
2   (date, start_time, end_time, capacity, predicted_traffic)
3 VALUES
4   -- Tomorrow slots
5   (CURRENT_DATE + 1, '06:00', '08:00', 10, 'NORMAL'),
6   (CURRENT_DATE + 1, '08:00', '10:00', 10, 'CONGESTED'),
7   (CURRENT_DATE + 1, '10:00', '12:00', 10, 'NORMAL'),
8   (CURRENT_DATE + 1, '12:00', '14:00', 10, 'NORMAL'),
9   (CURRENT_DATE + 1, '14:00', '16:00', 10, 'MODERATE'),
10  (CURRENT_DATE + 1, '16:00', '18:00', 10, 'CONGESTED'),
11  (CURRENT_DATE + 1, '18:00', '20:00', 10, 'MODERATE'),
12  (CURRENT_DATE + 1, '20:00', '22:00', 10, 'NORMAL'),
13  -- Day after tomorrow
14  (CURRENT_DATE + 2, '06:00', '08:00', 10, 'NORMAL'),
15  (CURRENT_DATE + 2, '08:00', '10:00', 10, 'CONGESTED'),
16  (CURRENT_DATE + 2, '10:00', '12:00', 10, 'NORMAL'),
17  (CURRENT_DATE + 2, '12:00', '14:00', 10, 'NORMAL'),
18  (CURRENT_DATE + 2, '14:00', '16:00', 10, 'MODERATE'),
19  (CURRENT_DATE + 2, '16:00', '18:00', 10, 'CONGESTED'),
20  (CURRENT_DATE + 2, '18:00', '20:00', 10, 'MODERATE'),
21  (CURRENT_DATE + 2, '20:00', '22:00', 10, 'NORMAL');

```

Listing 16: Demo Time Slots

### 3.4.2 Historical Traffic Data

```

1 -- Generate 7 days of historical traffic data
2 INSERT INTO traffic_updates
3   (camera_id, timestamp, vehicle_count, truck_count, status)
4 SELECT
5   'CAM_01_KING_ABDULAZIZ',
6   generate_series(
7     NOW() - INTERVAL '7 days',
8     NOW(),
9     INTERVAL '30 minutes'
10  ) as timestamp,
11  -- More traffic during rush hours (7-9 AM, 4-6 PM)
12  CASE
13    WHEN EXTRACT(HOUR FROM timestamp) BETWEEN 7 AND 9
14      THEN 140 + (random() * 30)::int
15    WHEN EXTRACT(HOUR FROM timestamp) BETWEEN 16 AND 18
16      THEN 135 + (random() * 25)::int

```

```

17     WHEN EXTRACT(HOUR FROM timestamp) BETWEEN 10 AND 15
18     THEN 80 + (random() * 40)::int
19     ELSE 60 + (random() * 30)::int
20 END as vehicle_count,
21 -- Trucks vary
22 (10 + (random() * 10))::int as truck_count,
23 -- Status based on count
24 CASE
25     WHEN vehicle_count > 150 THEN 'CONGESTED'
26     WHEN vehicle_count > 100 THEN 'MODERATE'
27     ELSE 'NORMAL'
28 END as status
29 FROM generate_series(1, 336);
30 -- 7 days * 48 half-hour intervals

```

Listing 17: Generate Historical Data for Demo

### 3.5 Database Views

```

1  -- Active permits with details
2  CREATE VIEW active_permits AS
3  SELECT
4      p.id, p.qr_code, p.status, p.priority, p.cargo_type,
5      d.name as driver_name,
6      d.phone as driver_phone,
7      d.vehicle_plate,
8      ts.date as slot_date,
9      ts.start_time,
10     ts.end_time,
11     pr.description as priority_description,
12     pr.max_delay_minutes,
13     pr.can_be_halted
14 FROM permits p
15 JOIN drivers d ON p.driver_id = d.id
16 JOIN time_slots ts ON p.slot_id = ts.id
17 LEFT JOIN priority_rules pr ON p.cargo_type = pr.cargo_type
18 WHERE p.status IN ('APPROVED', 'PENDING', 'HALTED');
19
20 -- Latest traffic per camera
21 CREATE VIEW latest_traffic AS
22 SELECT DISTINCT ON (camera_id)
23     camera_id, timestamp, vehicle_count,
24     truck_count, status, recommendation
25 FROM traffic_updates
26 ORDER BY camera_id, timestamp DESC;
27
28 -- Daily statistics
29 CREATE VIEW daily_stats AS
30 SELECT
31     DATE(created_at) as date,
32     COUNT(*) as total_permits,
33     COUNT(*) FILTER (WHERE status = 'HALTED')
34     as halted_count,
35     COUNT(*) FILTER (WHERE priority = 'EMERGENCY')
36     as emergency_count,

```

```
37     AVG(EXTRACT(EPOCH FROM (updated_at - created_at))/60)::int
38     as avg_delay_minutes
39 FROM permits
40 WHERE created_at > NOW() - INTERVAL '30 days'
41 GROUP BY DATE(created_at)
42 ORDER BY date DESC;
43
44 -- Vessel impact analysis
45 CREATE VIEW vessel_impact AS
46 SELECT
47     v.vessel_name,
48     v.arrival_date,
49     v.estimated_trucks,
50     COUNT(p.id) as actual_permits,
51     AVG(CASE
52         WHEN p.status = 'HALTED' THEN 1 ELSE 0
53     END) * 100 as halt_percentage
54 FROM vessel_schedules v
55 LEFT JOIN permits p ON p.vessel_id = v.id
56 GROUP BY v.id, v.vessel_name, v.arrival_date, v.estimated_trucks
57 ORDER BY v.arrival_date DESC;
```

Listing 18: Useful Views

## 4 Impact Analysis

### 4.1 Overview

This system manages 1,260 daily truck movements using a four-tier priority classification. Unlike traditional systems treating all trucks equally, this approach protects time-critical cargo while managing overall congestion through intelligent prioritization.

### 4.2 Priority Tier Framework

#### 4.2.1 Classification System

Priority	Cargo Types	Examples
EMERGENCY	Life-critical cargo	Medical supplies, vaccines, blood products, food expires <24h
ESSENTIAL	Time-sensitive cargo	E-commerce (Amazon/Noon), JIT manufacturing, scheduled retail deliveries, perishables 24-48h
NORMAL	Standard freight	General containers, regular cargo, stock replenishment
LOW	Non-urgent freight	Bulk materials, construction supplies, storage-bound goods

Table 3: Priority Classification Framework

#### 4.2.2 Implementation Approach

**Prototype (Current):**

The hackathon prototype uses a trust-based system:

- Driver self-selects cargo type from dropdown during booking
- Mobile app provides clear tier definitions and examples
- System displays warning about penalties for false claims
- Dashboard monitors for suspicious usage patterns

**Known Limitation - system abuse Risk:**

Without automated verification, drivers could falsely claim higher priorities to avoid delays. This is acknowledged as a prototype limitation suitable for demonstration and pilot testing.

**Mitigation Strategy:**

*Immediate (Prototype/Pilot Phase):*

- Clear in-app education about appropriate tier use
- Warning messages about penalties for false claims
- Dashboard alerts flag accounts with 100% ESSENTIAL claims

- Manual review of suspicious patterns
- Start with voluntary participants (self-selecting honest actors)

*Production Enhancement (Future Work):*

- API integration with major shippers (Amazon, Aramex, DHL)
- Require supporting documentation (tracking number, PO, manifest)
- Tiered pricing as economic deterrent (ESSENTIAL costs premium)
- Random spot checks at port gate with penalty enforcement
- Partnership with logistics companies for pre-verification

#### **Realistic Assessment:**

Research on honor systems shows 70-80% compliance when:

- Consequences are clearly communicated
- Monitoring is visible to users
- System starts with voluntary participants
- Penalties for abuse are enforced

The expected 20-30% system abuse rate is manageable during pilot phase and decreases to 5% with production verification systems.

## **4.3 Stakeholder Impact Analysis**

### **4.3.1 Emergency Logistics (EMERGENCY Tier)**

**Stakeholders:** Hospitals, pharmacies, blood banks, critical food suppliers

**Impact:**

- 100% protection - never halted regardless of congestion
- Zero delays for life-critical medical supplies
- Prevents healthcare supply chain disruption
- Eliminates food spoilage for time-critical perishables

**Volume:** Estimated 2% of daily permits (approximately 25 permits/day)

### **4.3.2 Commercial Time-Sensitive Operations (ESSENTIAL Tier)**

**Stakeholders:** E-commerce companies (Amazon, Noon, Jahez), JIT manufacturers (SABIC, automotive suppliers), major retailers (Danube, Tamimi, Carrefour)

**Critical Requirements:**

- Same-day/next-day delivery commitments
- JIT manufacturing (production stops without parts)
- Scheduled retail deliveries (narrow time windows)
- SLA penalties for missed deliveries (\$50-200 per incident)

**System Response:**

- Protected during moderate congestion (90% of cases)
- Max 2-hour delay in severe congestion (10% of cases)

- Priority rescheduling with immediate alternatives
- 98% on-time delivery protection overall

**Impact:**

- Maintains e-commerce delivery commitments
- Prevents manufacturing line stoppages (\$500K-2M/hour cost)
- Protects retail supply schedules
- Reduces SLA penalty costs

**Volume:** Estimated 18% of daily permits (approximately 227 permits/day)

**Trust-Based Challenge:** Requires accurate self-classification. False ESSENTIAL claims by non-urgent cargo would dilute protection. Production verification addresses this through documentation requirements and API integration with shippers.

#### 4.3.3 Standard Freight Operations (NORMAL Tier)

**Stakeholders:** General trucking companies, standard container freight, regular cargo operations

**Impact:**

- 15-20% rescheduling rate during peak congestion periods
- Average delay: 2-4 hours with smart alternative suggestions
- Fuel savings: 20% from reduced idling and waiting
- Better schedule predictability overall
- One-tap rescheduling vs manual coordination

**Volume:** Estimated 65% of daily permits (approximately 819 permits/day)

#### 4.3.4 Bulk Materials & Non-Urgent Freight (LOW Tier)

**Stakeholders:** Construction logistics, bulk material suppliers, empty container returns, storage-bound goods

**Impact:**

- 30-40% rescheduling rate (highest flexibility)
- Average delay: 4-8 hours, up to 24 hours possible
- Incentivized off-peak scheduling
- Reduced permit costs (future tiered pricing)
- Requires flexible logistics planning

**Volume:** Estimated 15% of daily permits (approximately 189 permits/day)

#### 4.3.5 Port Operations & Maritime Logistics

**Stakeholders:** Dammam Port Authority, shipping lines (Maersk, MSC, Evergreen), freight forwarders, customs brokers

**Positive Impacts:**

- 25% reduction in port gate congestion

- More predictable truck arrival patterns
- Vessel schedule coordination reduces cargo dwell time
- Better dock utilization through managed flow
- Reduced demurrage costs through priority protection

**Challenges:**

- Cargo availability doesn't always match permit schedules
- Customs clearance delays create bottlenecks
- Multiple trucking companies per shipment coordination
- Container release timing variability

**Mitigation:**

- Vessel schedule display helps anticipate surges
- Flexible permit adjustment for operational delays
- Emergency override for vessel schedule dependencies
- Future: Real-time Mawani API integration (production phase)

#### 4.3.6 Trucking Companies & Drivers

**Stakeholders:** 1,260 daily truck movements representing 300-500 companies and independent drivers

**Positive Impacts:**

- Predictable scheduling reduces total trip time by 20%
- 20% fuel savings from reduced idling and waiting
- Better route planning with traffic predictions
- One-tap rescheduling vs manual coordination
- Priority access for time-sensitive loads
- Clearer customer communication (show permit status)

**Negative Impacts:**

- NORMAL/LOW priority faces 15-30% rescheduling rate
- Potential revenue loss from delayed deliveries
- Learning curve for new system (1-2 weeks typical)
- Smartphone and mobile app literacy requirements
- Dependency on reliable mobile connectivity

**Adaptation Strategy:**

- Gradual rollout with voluntary early adopters
- In-app tutorials and support hotline
- Driver training sessions at trucking associations
- Grace period with manual overrides during transition

#### 4.3.7 Commuters & Residents

**Stakeholders:** 50,000+ daily commuters on King Abdulaziz Road, residential neighborhoods

**Positive Impacts:**

- 30% congestion reduction during peak hours (7-9 AM, 4-6 PM)
- 25% average commute time improvement
- 15% accident reduction through better traffic flow
- Improved air quality from reduced emissions
- Better quality of life in residential areas
- Safer school routes

**Environmental Benefits:**

- 8,200 tons CO2 reduction annually
- 15-20% reduction in vehicle emissions
- 40% reduction in idling time
- Noise pollution reduction during evening hours

### 4.4 Risk Analysis

#### 4.4.1 System Abuse Risk

**Risk:** Drivers falsely claiming ESSENTIAL/EMERGENCY priority to avoid delays

**Likelihood:** Medium-High during initial deployment without verification

**Impact:** If ~50% falsely claim ESSENTIAL, system effectiveness degrades significantly

**Consequence Cascade:**

- True emergency cargo loses protection
- Commercial time-sensitive operations face delays
- System credibility damaged
- Stakeholder trust eroded
- Return to baseline congestion levels

**Mitigation - Phased Approach:**

*Phase 1 (Months 1-3): Trust-Based Pilot*

- Voluntary participants (self-selecting honest actors)
- Clear education about consequences
- Dashboard monitoring for suspicious patterns
- Manual review and warnings for obvious abuse
- Gather data on actual system abuse rates

*Phase 2 (Months 4-6): Soft Verification*

- Require description/justification for ESSENTIAL claims
- Spot checks at port gate (5-10% sample)
- Warnings and education for questionable claims

- Build verification partnerships with major shippers

*Phase 3 (Months 7-12): Full Verification*

- API integration with shipping companies
- Supporting documentation required (tracking, PO, manifest)
- Tiered pricing implemented (economic deterrent)
- Penalties enforced for confirmed false claims
- Random spot checks with consequence enforcement

#### 4.4.2 Commercial Liability Risk

**Risk:** System delays cause commercial SLA breaches

**Examples:**

- Amazon Prime delivery missed → \$150 penalty + customer satisfaction impact
- JIT parts delayed → \$500K-2M production line stoppage
- Retail delivery missed → Stock-out, revenue loss

**Mitigation:**

- ESSENTIAL tier provides 98% on-time protection
- 4-hour advance warning for predicted delays
- Emergency override capability for critical situations
- Clear liability framework (production deployment)
- Insurance coverage for system-caused failures (production)
- Documented SLA guarantees for verified ESSENTIAL cargo

**Legal Framework (Production):**

- System operates as traffic management, not logistics guarantee
- Force majeure clauses for extreme congestion
- Liability caps for system failures
- Dispute resolution process
- Insurance partnership for commercial claims

#### 4.4.3 Technical Risks

Risk	Impact	Mitigation
AI accuracy <90%	Incorrect traffic status	Multi-camera validation, human review
System downtime	No permit management	Redundant infrastructure, manual fallback
Network loss	Driver can't receive updates	Edge computing, SMS backup notifications
Database failure	Permit data loss	Automated backups, replication

Table 4: Technical Risk Mitigation

## 4.5 Cost-Benefit Analysis

### 4.5.1 Implementation Costs

#### Prototype (Current):

- Development: Student labor (hackathon - 5 days)
- Infrastructure: Free tier services (Supabase, Vercel, Expo)
- Demo equipment: Existing cameras and computers
- **Total Cost: Near-zero**

#### Pilot Deployment (3-6 months):

- 2-3 traffic cameras: \$20K-40K
- Basic backend infrastructure: \$10K-20K
- Support team (part-time): \$30K-50K
- **Total Cost: \$60K-110K**

#### Production Deployment (Full Scale):

- 10-30 AI cameras: \$200K-400K
- Edge computing infrastructure: \$50K-100K
- Backend servers & database: \$40K-60K
- Network connectivity: \$20K-40K
- Initial development & testing: \$50K-100K
- 24/7 operations team: \$180K-250K annually
- **Total CAPEX: \$540K-950K**
- **Annual OPEX: \$250K-400K**

### 4.5.2 Economic Benefits

#### Quantifiable Annual Savings:

##### Return on Investment:

- Benefits: 38.3M SAR annually
- Costs: 0.95M SAR (CAPEX + first year OPEX)
- ROI: 40:1 benefit-to-cost ratio
- Payback period: 2-3 months

##### Priority-Specific Economic Value:

- EMERGENCY: 2M SAR saved in healthcare delays, spoilage prevention
- ESSENTIAL: 5M SAR saved in SLA penalties, manufacturing downtime
- NORMAL/LOW: 3M SAR saved in fuel, driver wages
- Commuters: 6M SAR saved in time and fuel
- Port operations: 12M SAR saved in demurrage reduction

Benefit Category	Annual Value (SAR)	Calculation Basis
Truck fuel savings	4.2M	1,260 trucks × 20% fuel × 45 SAR/day × 365 days
Commuter fuel savings	2.8M	10,000 cars × 15% fuel × 5 SAR/day × 365 days
Time savings (trucks)	8.1M	1,260 trucks × 1.5h saved × 150 SAR/h × 365 days
Accident reduction	3.5M	15% fewer accidents × average cost
Demurrage reduction	12.0M	80% reduction in port delay penalties
Productivity gains	6.5M	Faster deliveries, less driver downtime
Environmental (carbon)	1.2M	CO2 reduction monetization
<b>Total Annual Benefits</b>	<b>38.3M SAR</b>	<b>(\$10.2M USD)</b>

Table 5: Annual Economic Benefits

## 4.6 Social & Environmental Impact

### 4.6.1 Quality of Life Improvements

- 25% commute time reduction for 50,000+ daily commuters
- Safer school routes (15% accident reduction)
- Reduced evening noise pollution (trucks avoid residential peak hours)
- Better air quality in residential neighborhoods
- Increased property values near previously congested zones
- Reduced stress and improved well-being for residents

### 4.6.2 Environmental Benefits

Metric	Annual Reduction
CO2 emissions	8,200 tons
Fuel consumption	3.2 million liters
Particulate matter (PM2.5)	15-18% reduction
NOx emissions	18-22% reduction
Vehicle idling time	40% reduction
Total vehicle-hours in congestion	35% reduction

Table 6: Environmental Impact Metrics

#### Vision 2030 Alignment:

- Green logistics initiative support
- Sustainable urban development
- Climate action commitments
- Smart city digital transformation
- Quality of life improvements

## 4.7 Conclusion

The priority-aware traffic management system demonstrates strong positive impact across stakeholder groups while maintaining realistic expectations about prototype limitations.

### **Key Strengths:**

- Protects critical cargo (100% EMERGENCY, 98% ESSENTIAL)
- Addresses commercial needs (e-commerce, JIT, retail)
- 30% congestion reduction benefits all road users
- 40:1 benefit-to-cost ratio
- Simple trust-based prototype enables rapid testing
- Clear path to production verification systems

### **Acknowledged Limitations:**

- Trust-based system vulnerable to 20-30% system abuse rate
- No automated commercial verification in prototype
- Limited enforcement capability during pilot phase
- Requires stakeholder education and buy-in

### **Critical Success Factors:**

- Start with voluntary pilot participants
- Clear communication about appropriate priority use
- Visible monitoring deters system abuse
- Phased verification implementation based on observed behavior
- Strong partnerships with port, shippers, and trucking associations
- Gradual rollout allows system refinement

With honest acknowledgment of limitations, phased deployment approach, and clear path to production verification, the system offers compelling value proposition for all stakeholders while maintaining realistic implementation timeline.

## 5 Mitigation Strategies

### 5.1 Priority Tier System

Priority	Examples	Treatment
EMERGENCY	Medical supplies	Never halted
ESSENTIAL	Food, fuel	Extreme only
NORMAL	Regular	Standard rules
LOW	Non-urgent	First rescheduled

### 5.2 Predictive Warnings

- 2h before: "85% congestion risk"
- 1h before: "Consider 10 AM instead"
- 30m before: "Final warning"

### 5.3 Compensation

#### 1. Reimbursement Fund

- Portion of fees
- Provable losses
- Liberal first 6 months

#### 2. Credits

- 3 reschedules = free slot
- Build loyalty

#### 3. Dynamic Pricing

- 8 AM risky = 500 SAR
- 2 PM clear = 200 SAR
- Market forces

### 5.4 Alternative Routes

```

1 {
2   "status": "HALTED",
3   "reason": "King Abdulaziz Road congested",
4   "alternatives": [
5     {"route": "King Fahd Road", "delay": "+15 min"},
6     {"route": "Eastern Ring Road", "delay": "+25 min"}
7   ],
8   "next_slot": "10:00 AM"
9 }
```

Listing 19: Alternative Response

## 5.5 Priority System Abuse Mitigation

### 5.5.1 Challenge

The trust-based priority selection system is vulnerable to system abuse, where drivers falsely claim higher priorities (ESSENTIAL/EMERGENCY) to avoid delays.

### 5.5.2 Why This Matters

If system abuse becomes widespread (>50% false claims):

- True emergency cargo loses protection
- System effectiveness degrades to baseline
- Stakeholder trust is damaged
- Commercial time-sensitive operations face unexpected delays

### 5.5.3 Mitigation Strategies

#### Education & Transparency

- Clear in-app definitions with examples for each tier
- Prominent warnings about penalties for false claims
- Dashboard showing system-wide usage statistics (creates accountability)
- Regular communication about why accurate classification matters

#### Pattern Monitoring

- Dashboard alerts for suspicious patterns:
  - Accounts claiming 100% ESSENTIAL priority
  - Sudden shifts in priority distribution
  - Unusual timing patterns
- Manual review and warning messages for questionable accounts
- Aggregate statistics visible to all users (social pressure)

#### Gradual Verification Implementation Phase 1 (Months 1-3): Trust + Monitoring

- Honor system with clear education
- Pattern detection and warnings
- Voluntary participants (self-selecting honest actors)
- Gather baseline system abuse rate data

#### Phase 2 (Months 4-6): Soft Verification

- Require brief description for ESSENTIAL claims
- Spot checks at port gate (5-10% sample)
- Educational feedback for questionable claims
- Build partnerships with major shippers

#### Phase 3 (Months 7+): Full Verification

- API integration with shipping companies
- Supporting documentation required (tracking , PO, manifest)
- Tiered pricing (ESSENTIAL costs premium - economic deterrent)
- Random spot checks with penalty enforcement
- Verified company pre-registration (Amazon, DHL, etc.)

Phase	Violation Response	Enforcement
Pilot (1-3 months)	Warning message, education	Soft (learning phase)
Transition (4-6 months)	Account review, temporary restrictions	Medium
Production (7+ months)	Fines (5K-10K SAR), suspension	Strict

Table 7: Phased Enforcement Approach

## Enforcement Framework

### 5.5.4 Expected Outcomes

#### Based on Honor System Research:

- Phase 1: 20-30% system abuse rate (manageable with monitoring)
- Phase 2: 10-15% system abuse rate (soft verification improves compliance)
- Phase 3: 5% system abuse rate (full verification nearly eliminates abuse)

#### Success Indicators:

- Emergency cargo maintains 100% on-time rate
- ESSENTIAL tier stays under 25% of total permits
- No widespread complaints about system abuse
- Commercial partners report 98%+ reliability

### 5.5.5 Contingency Plan

If system abuse exceeds 40% during pilot:

- Accelerate verification implementation timeline
- Temporarily increase spot check frequency to 20-30%
- Launch targeted education campaign
- Engage trucking associations for peer accountability
- Consider early introduction of tiered pricing

## 5.6 Gradual Rollout

### Phased Implementation

**Phase 1 (Months 1-2):** Monitoring only, no enforcement

**Phase 2 (Months 3-4):** Soft enforcement, 2h notice, manual overrides

**Phase 3 (Months 5+):** Full enforcement with all mitigations

## 5.7 Stakeholder Engagement

### **Before:**

- Meet trucking companies
- Driver focus groups
- Voluntary pilot
- Transparent data

### **Ongoing:**

- Monthly surveys
- Quarterly meetings
- Public dashboard
- Dispute ombudsman

## 6 Success Metrics

### 6.1 Traffic Metrics

Metric	Target	Measure
Peak Congestion	-30%	Avg count 7-9 AM
Truck Count	-40%	Trucks/hour peak
Accidents	-15%	Per 10K vehicles
Speed	+25%	Mean peak speed

### 6.2 Driver Metrics

Metric	Target
Rescheduling Rate	↓20% of permits
Delivery Time	-20% reduction
Satisfaction	↑70% positive
App Adoption	↑85% active

### 6.3 System Performance

Metric	Target
Uptime	↑99.5%
AI Accuracy	↑90%
API Response	↓500ms
Notification	↑95% delivery
GPS Update	Every 30s

### 6.4 Enhanced Features Metrics

### 6.5 Port Operations Metrics

Metric	Target	Measure
Emergency Cargo Protection	100%	Never halted or delayed
Essential Cargo Delay	≤2 hours	Max delay for urgent cargo
Prediction Accuracy	≥80%	Correct status prediction
Reschedule Acceptance	≥70%	Drivers accepting alternatives
Average Reschedule Time	≤2 min	Time to suggest alternatives
Priority Compliance	100%	Respect tier rules

Table 8: Enhanced System Success Metrics

Metric	Target	Measure
Average Dwell Time	-25%	Time truck spends in port area
On-Time Cargo Delivery	≥90%	Against shipping schedules
Vessel Coordination	100%	Tracked arrivals
Port Throughput	+15%	Containers/trucks per day

Table 9: Port Logistics Performance Indicators

## 7 Future Work & Enhancement Roadmap

### 7.1 Prototype vs Production Context

#### 7.1.1 Current Hackathon Prototype Scope

This demonstration system focuses on proving core concepts:

##### Prototype Demonstrates

- AI-powered traffic detection with YOLO
- Four-tier priority classification logic
- Real-time dynamic permit management
- Predictive traffic analytics
- Mobile driver application with smart rescheduling
- Administrative dashboard with vessel schedules
- Complete database architecture

### 7.1.2 Prototype Limitations

#### Known Limitations

- Trust-based priority selection (no automated verification)
- Simulated vessel schedule data (no real Mawani API)
- Heuristic traffic prediction (not ML models)
- Single camera demonstration footage
- Free tier infrastructure (not production-grade)
- No 24/7 operational support

### 7.1.3 Production Deployment Requirements

Full operational deployment requires significant enhancements beyond hackathon scope:

Category	Prototype	Production Needed
Priority Verification	Trust-based	API integration, documentation checks
Mawani Integration	Simulated data	Official API access, real-time sync
Traffic Prediction	Heuristic rules	ML models, historical training
Camera Network	1 demo camera	10-30 cameras with edge computing
Infrastructure	Free tiers	Enterprise hosting, redundancy
Support	None	24/7 monitoring team
Legal Framework	N/A	Liability agreements, insurance

Table 10: Prototype vs Production Requirements

## 7.2 Short-Term (6-12 months)

### 1. Multi-Camera

- 5-10 cameras
- Aggregate data
- Identify bottlenecks

### 2. Mobile Improvements

- In-app chat
- History analytics
- Favorites
- Community

### 3. Admin Tools

- Manual overrides
- Bulk management
- Advanced analytics
- Audit logs

## 7.3 Medium-Term (1-2 years)

### 1. Predictive Analytics

- ML on historical data
- 24h forecasts
- Pattern recognition
- Event adjustments

### 2. Multi-City

- Riyadh, Jeddah, Mecca
- Shared accounts
- Regional coordination

### 3. Integration

- Traffic Department
- Customs systems
- Fleet software
- Navigation apps

## 7.4 Long-Term (2+ years)

### 1. Autonomous Vehicles

- Self-driving API
- V2X communication
- Platoon management

### 2. Smart City

- Traffic light coordination
- Dynamic lanes
- Holistic mobility

### 3. Blockchain

- Immutable records
- Smart contracts
- Dispute resolution

## 8 Conclusion

### 8.1 Executive Summary

The Smart Logistics & Truck Management System represents a data-driven approach to solving port-induced urban traffic congestion through AI-powered dynamic permit management with intelligent priority handling. Developed in response to the Dammam Urban Development Hackathon 2025 (Challenge #2), this system addresses the critical intersection of maritime logistics and urban mobility on King Abdulaziz Road.

With approximately 1,260 daily truck movements serving Dammam seaport, this solution balances the competing demands of efficient port operations, commuter traffic flow, and residential quality of life. By implementing priority-aware management, predictive analytics, and intelligent rescheduling, the system ensures critical cargo maintains flow while managing overall congestion.

### 8.2 Key Takeaways

Benefits
<ul style="list-style-type: none"><li>• 30-40% congestion reduction through smart scheduling</li><li>• 100% protection for emergency cargo (medical, perishable)</li><li>• 15-20% accident reduction</li><li>• 25% delivery time improvement</li><li>• 80%+ traffic prediction accuracy</li><li>• Millions in savings from reduced fuel waste and delays</li><li>• Vision 2030 alignment for digital transformation</li></ul>
Challenges
<ul style="list-style-type: none"><li>• Logistics disruption during adoption phase</li><li>• \$150K-250K pilot cost (10 cameras, 3 months)</li><li>• Reliability critical for port operations</li><li>• Stakeholder buy-in from trucking companies</li><li>• 12-18 month stabilization period</li><li>• Eventual Mawani API integration required</li></ul>

### 8.3 Success Factors

1. Gradual rollout with voluntary pilot phase
2. Strong partnerships with port authority and trucking associations
3. Robust mitigation strategies for affected drivers
4. Technical excellence in AI accuracy and system reliability
5. Continuous improvement based on real-world data
6. Clear communication about priority tier rationale

## 8.4 Hackathon Prototype vs Production System

### 8.4.1 Current Prototype Scope

This hackathon prototype demonstrates core concepts using simulated data and focuses on proving feasibility:

- AI-powered traffic detection and classification
- Priority-based permit management logic
- Predictive traffic analytics
- Mobile driver application with rescheduling
- Real-time dashboard with vessel schedules
- Complete database architecture

### 8.4.2 Production Deployment Requirements

Full deployment would require:

#### 1. Government Integration

- Official Mawani API access and partnership
- Saudi Ports Authority coordination
- Ministry of Transport approval
- Customs system integration

#### 2. Infrastructure

- 10-30 high-resolution traffic cameras
- Edge computing for real-time processing
- Redundant server infrastructure
- 24/7 monitoring and support

#### 3. Legal Framework

- Liability agreements for system-caused delays
- Data privacy compliance
- Driver consent mechanisms
- Dispute resolution procedures

#### 4. Stakeholder Engagement

- Trucking association partnerships
- Driver training programs
- Port operator coordination
- Public communication campaigns

Phase	Duration	Scope
Phase 1	3 months	Prototype pilot with 2-3 cameras, voluntary participation
Phase 2	6 months	Expand to 10 cameras, soft enforcement, manual overrides available
Phase 3	12 months	Full system with Mawani integration, all features operational

#### 8.4.3 Phased Implementation Timeline

### 8.5 Final Recommendation

**Build this system** with careful execution and stakeholder collaboration. View as a sociotechnical transformation that balances:

- Urban mobility and resident quality of life
- Port efficiency and maritime commerce
- Economic activity and business operations
- Environmental sustainability
- Driver livelihoods and working conditions
- Public trust in AI-driven governance

This project represents a blueprint for 21st-century smart port cities where AI and humans work together to create more efficient, sustainable, and livable urban environments. The priority-aware approach ensures that technology serves human needs - protecting critical cargo while managing traffic flow intelligently.

By aligning with Saudi Vision 2030's goals for digital transformation, logistics excellence, and quality of life improvements, this system positions Dammam as a model for smart port cities in the region and globally.

*A smarter road to the future.*

## Appendix A: API Documentation

### Traffic Update

#### POST /api/traffic

```
1 {  
2   "camera_id": "CAM_01_KING_ABDULAZIZ",  
3   "timestamp": "2026-02-05T20:30:00Z",  
4   "status": "CONGESTED",  
5   "vehicle_count": 145,  
6   "truck_count": 12,  
7   "recommendation": "HALT_TRUCK_PERMITS"  
8 }
```

Listing 20: Request

```
1 {  
2   "success": true,  
3   "permits_affected": 5,  
4   "permits_protected": 2  
5 }
```

Listing 21: Response

### Get Time Slots

#### GET /api/slots?date=YYYY-MM-DD

```
1 {  
2   "slots": [  
3     {  
4       "id": "uuid",  
5       "date": "2026-02-06",  
6       "start_time": "08:00",  
7       "end_time": "10:00",  
8       "capacity": 10,  
9       "booked": 7,  
10      "status": "AVAILABLE",  
11      "predicted_traffic": "MODERATE"  
12    }  
13  ]  
14 }
```

Listing 22: Response

### Book Permit

#### POST /api/book

```
1 {  
2   "slot_id": "uuid",  
3   "priority": "NORMAL",  
4   "cargo_type": "STANDARD"  
5 }
```

Listing 23: Request

```
1 {
2   "permit": {
3     "id": "uuid",
4     "qr_code": "PERMIT-2026-001",
5     "status": "APPROVED",
6     "priority": "NORMAL",
7     "cargo_type": "STANDARD",
8     "slot_date": "2026-02-06",
9     "start_time": "08:00",
10    "can_be_halted": true,
11    "max_delay_minutes": 480
12  }
13 }
```

Listing 24: Response

## Get Vessel Schedule

GET /api/vessels?date=YYYY-MM-DD

```
1 {
2   "vessels": [
3     {
4       "id": "uuid",
5       "vessel_name": "MSC TOKYO",
6       "arrival_date": "2026-02-06",
7       "arrival_time": "08:00",
8       "estimated_trucks": 180,
9       "actual_trucks": 0,
10      "status": "SCHEDULED",
11      "cargo_priority": "NORMAL"
12    }
13  ]
14 }
```

Listing 25: Response

## Traffic Prediction

GET /api/predict?hours=1&camera\_id=CAM\_01

```
1 {
2   "predicted_status": "MODERATE",
3   "confidence": 82.5,
4   "predicted_for": "2026-02-06T15:00:00Z",
5   "factors": {
6     "rush_hour": false,
7     "vessel_arrivals": 2,
8     "weekend": false,
9     "historical_pattern": "MODERATE",
10    "current_trend": "INCREASING"
11  },
12   "recommendation": "Reduce new permit approvals after 4:30 PM"
13 }
```

Listing 26: Response

## Get Alternative Slots

GET /api/slots/alternatives?permit\_id=uuid

```
1 {
2   "original_slot": {
3     "date": "2026-02-06",
4     "start_time": "08:00",
5     "end_time": "10:00"
6   },
7   "alternatives": [
8     {
9       "slot_id": "uuid",
10      "date": "2026-02-06",
11      "start_time": "10:00",
12      "end_time": "12:00",
13      "available_spots": 8,
14      "predicted_traffic": "NORMAL",
15      "confidence": 85,
16      "recommended": true
17    },
18    {
19      "slot_id": "uuid",
20      "date": "2026-02-06",
21      "start_time": "14:00",
22      "end_time": "16:00",
23      "available_spots": 6,
24      "predicted_traffic": "MODERATE",
25      "confidence": 78,
26      "recommended": false
27    }
28  ]
29 }
```

Listing 27: Response

## Reschedule Permit

POST /api/permits/reschedule

```
1 {
2   "permit_id": "uuid",
3   "new_slot_id": "uuid"
4 }
```

Listing 28: Request

```
1 {
2   "success": true,
3   "permit": {
4     "id": "uuid",
5     "status": "APPROVED",
6     "new_slot_date": "2026-02-06",
7     "new_start_time": "10:00",
8     "rescheduled_count": 1
9   }
}
```

10 }

Listing 29: Response

## Get Daily Analytics

GET /api/analytics/daily?days=7

```
1 {
2   "period": {
3     "start_date": "2026-01-30",
4     "end_date": "2026-02-06",
5     "days": 7
6   },
7   "summary": {
8     "total_permits": 245,
9     "approved_permits": 210,
10    "halted_permits": 28,
11    "emergency_permits": 7,
12    "avg_delay_minutes": 45,
13    "reschedule_rate": 11.4
14  },
15  "daily_breakdown": [
16    {
17      "date": "2026-02-06",
18      "total_permits": 38,
19      "halted_count": 5,
20      "emergency_count": 1,
21      "avg_delay_minutes": 52
22    }
23  ]
24 }
```

Listing 30: Response

## Update Driver Location

POST /api/locations

```
1 {
2   "permit_id": "uuid",
3   "latitude": 26.3927,
4   "longitude": 50.0772,
5   "accuracy": 12.5,
6   "speed": 45.2,
7   "eta_minutes": 15
8 }
```

Listing 31: Request

```
1 {
2   "success": true,
3   "message": "Location updated"
4 }
```

Listing 32: Response

## Get Priority Rules

GET /api/priority-rules

```
1 {  
2   "rules": [  
3     {  
4       "cargo_type": "PERISHABLE",  
5       "priority_level": "EMERGENCY",  
6       "max_delay_minutes": 0,  
7       "can_be_halted": false,  
8       "description": "Food, medicine - expires within hours",  
9       "color_code": "#EF4444"  
10    },  
11    {  
12      "cargo_type": "MEDICAL",  
13      "priority_level": "EMERGENCY",  
14      "max_delay_minutes": 0,  
15      "can_be_halted": false,  
16      "description": "Medical supplies, vaccines",  
17      "color_code": "#EF4444"  
18    }  
19  ]  
20 }
```

Listing 33: Response

## Appendix B: Collaboration Guidelines

### Communication

#### 1. Daily Standup (15 min, 9 AM)

- Yesterday's work
- Today's plan
- Blockers

#### 2. Progress Tracker

- Google Sheets/Notion
- Real-time updates
- Flag blockers

#### 3. Integration Test (30 min, 8 PM)

- All systems together
- Verify API
- Catch changes

### Git Workflow

#### Branches:

- main - Production
- dev - Integration
- feature/name - Features

#### Commits:

[Team X] Brief description

- Detailed change 1
- Detailed change 2

### Testing

**Team 1:** API unit tests, DB integration, E2E dashboard

**Team 2:** Component tests, navigation, notifications

**Team 3:** Classification unit tests, mock API, accuracy

**All:** Weekly full system test

## Appendix C: Glossary

**AI** Artificial Intelligence

**API** Application Programming Interface

**COCO** Common Objects in Context dataset

**Demurrage** Penalty fees charged when cargo remains at port beyond free time

**ETA** Estimated Time of Arrival

**Expo** React Native development framework

**GPS** Global Positioning System

**JIT** Just-In-Time (manufacturing/delivery strategy)

**JSON** JavaScript Object Notation

**JWT** JSON Web Token

**Mawani** Saudi Ports Authority ()

**Mina Road** Port Road connecting to Dammam seaport

**OTP** One-Time Password

**PostgreSQL** Relational database management system

**QR Code** Quick Response barcode

**REST** Representational State Transfer

**RLS** Row Level Security

**RTL** Right-to-Left text direction

**SSE** Server-Sent Events

**Supabase** Open-source Firebase alternative

**TEU** Twenty-foot Equivalent Unit (container measurement)

**YOLO** You Only Look Once object detection algorithm

**Zod** TypeScript-first schema validation library

**Zustand** Lightweight state management library

## Priority Tier Definitions

**EMERGENCY** Highest priority - cargo that cannot be delayed (medical supplies, perishables)

**ESSENTIAL** High priority - time-sensitive cargo with 2-hour max delay tolerance

**NORMAL** Standard priority - regular freight with 8-hour flexibility

**LOW** Lowest priority - bulk materials that can wait up to 24 hours

## Traffic Status Levels

**NORMAL** Vehicle count below 100, free-flowing traffic

**MODERATE** Vehicle count 100-150, slower movement, some delays

**CONGESTED** Vehicle count above 150, significant delays, permits may be halted

## Appendix D: Resources

### Team Leads

**Team 1:** Mohammad Al-Sadah **Team 2:**  
Amjad Al-Leef **Team 3:**  
Mohammad Al Rabeh, Ali Al Baqqal, Hassan Al Zourei

### Repositories

- Team 1: [github.com/MOHKSADAH/smart-logistics-web](https://github.com/MOHKSADAH/smart-logistics-web)
- Team 2: [github.com/depends on the user/driver-mobile-app](https://github.com/depends-on-the-user/driver-mobile-app)
- Team 3: [github.com/depends on the user/traffic-vision-ai](https://github.com/depends-on-the-user/traffic-vision-ai)

### Documentation

- Next.js: [nextjs.org/docs](https://nextjs.org/docs)
- Expo: [docs.expo.dev](https://docs.expo.dev)
- YOLO: [docs.ultralytics.com](https://docs.ultralytics.com)
- Supabase: [supabase.com/docs](https://supabase.com/docs)