Functional Specification Document
Smart Urban Mobility Project
NEURONIX Team

Djamel Eddine Dib
Amine Mazari Abdessameud
Akram Kaid
Abderrahmane Khoudour

April 25, 2025

INNOVERSE Hackathon 2025, Oran

Contents

1 Introduction

This document outlines the Software Requirements Specification (SRS) for the Minimum Viable Product (MVP) of the Smart Urban Mobility application, developed by the NEURONIX team for the INNOVERSE Hackathon 2025 in Oran, Algeria. The project leverages Artificial Intelligence (AI) and Internet of Things (IoT) technologies to enhance urban transportation efficiency, safety, and accessibility in the city of Oran. It adheres to the IEEE 830-1998 standard for software requirements specifications, ensuring clarity and completeness for all stakeholders.

The Smart Urban Mobility system comprises three core modules:

1. **Intelligent Traffic Signal System (ITSS)**: Optimizes traffic flow and enhances safety through AI-driven traffic signal control, emergency vehicle prioritization, and real-time traffic violation detection.

2. **Smart Predictive Maintenance System (SPMS)**: Monitors the health of public transport infrastructure (e.g.,railways, roads) using IoT sensors and AI algorithms to predict maintenance needs.

3. **Smart Parking and Transit System (SPT)**: Improves parking efficiency with real-time availability and guidance, and optimizes bus line planning and scheduling.

This SRS serves as the primary reference for the NEURONIX team, hackathon organizers, and stakeholders, ensuring alignment on the system's objectives and capabilities.

1.1 Purpose

The purpose of this SRS is to define the functional and non-functional requirements for the Smart Urban Mobility MVP, providing a clear roadmap for development and evaluation during the INNOVERSE Hackathon 2025. It specifies the system's features, performance criteria, security measures, and technical constraints, facilitating collaboration among developers, designers, testers, and stakeholders. The document aims to ensure that the MVP addresses Oran's transportation challenges, including congestion, inefficient traffic control, and limited transit accessibility, while laying the foundation for a scalable smart city solution.

1.2 Scope

The Smart Urban Mobility MVP is an integrated software and hardware solution designed to modernize transportation in Oran, targeting city administrators, traffic controllers, public transport operators, and citizens. The system includes:

- **Intelligent Traffic Signal System (ITSS)**:
  - AI-powered adaptive traffic signal control leveraging real-time traffic density analysis. This system employs Convolutional Neural Networks (CNNs) for precise vehicle detection, classification (e.g., cars, trucks, motorcycles), and queue length estimation from high-resolution camera feeds. The CNN output that dynamically adjusts signal timings (green light duration, phase sequence) to minimize overall network delay, maximize throughput, and improve pedestrian safety.
  - Emergency vehicle prioritization via CNN deep learning algorithms and camera-based detection (e.g., Faster R-CNN).
  - Real-time traffic violation detection (e.g., red-light running) using YOLOv8 and Optical Character Recognition (OCR) for license plates.
  - Accident detection with automated alerts and dynamic signage (e.g., Panneau Triflash).

- **Smart Predictive Maintenance System (SPMS)**:
  - IoT sensors (e.g., vibration, thermal) to monitor infrastructure health (railways, roads).
  - AI models (e.g., LSTM) to predict maintenance needs and prevent failures.
  - Real-time passenger notifications for bus delays, rerouting, via the mobile app.
- **Transit System and Smart Parking (TSP)**:
  - **Bus Line Planning and Scheduling**:
    - GPS trackers for real-time bus location and capacity monitoring.
    - AI-driven route optimization to adjust schedules based on traffic and passenger demand.
    - Mobile app features for live bus tracking, estimated arrival times, delay alerts, and route planning.
  - **Parking Management**:
    - cameras to detect parking space occupancy.
    - Mobile app guidance to available spots with GPS integration (e.g., Google Maps).
    - Signage boards at intersections displaying parking availability.
- **Data Management**: Secure storage and processing of traffic, transit, and parking data with role-based access control and compliance with data protection regulations.
- **Notifications**: Real-time alerts for traffic incidents, bus delays, maintenance warnings, and parking updates via in-app messages or email.
- **Data Export**: Reports (e.g., traffic patterns, bus performance, parking usage) in PDF or CSV formats.

Out of Scope for the MVP:

- Integration with national transportation databases or external city systems.
- Autonomous vehicle coordination or vehicle-to-vehicle communication.
- Environmental monitoring (e.g., air quality, emissions).
- Multilingual interfaces (English only for MVP).

The MVP prioritizes a functional prototype for the hackathon, with scalability for future enhancements.

1.3 Definitions, Acronyms, and Abbreviations

- **Admin**: City administrator or traffic controller managing the system.
- **AI**: Artificial Intelligence – Machine learning models (e.g., CNNs, LSTM) for traffic, maintenance, and transit optimization.
- **API**: Application Programming Interface – RESTful JSON interface for system communication.
- **CNN**: Convolutional Neural Network – AI model for traffic and vehicle detection.
- **GPS**: Global Positioning System – Used for bus tracking.
- **IoT**: Internet of Things – Sensors for vehicle, infrastructure, and parking monitoring.

- **ITSS**: Intelligent Traffic Signal System – Traffic management module.
- **LSTM**: Long Short-Term Memory – AI model for predictive maintenance.
- **MQTT**: Message Queuing Telemetry Transport – Protocol for IoT data transmission.
- **OCR**: Optical Character Recognition – Technology for license plate detection.
- **RESTful**: Representational State Transfer – API architecture style.
- **SPS**: Smart Parking and Transit System – Module for parking and bus management.
- **SPMS**: Smart Predictive Maintenance System – Maintenance module.
- **SRS**: Software Requirements Specification – This document.
- **TLS**: Transport Layer Security – Protocol for secure data transmission.
- **UML**: Unified Modeling Language – For system modeling.
- **YOLO**: You Only Look Once – AI model for real-time object detection.

1.4 Overview

This SRS is organized per IEEE 830-1998 to provide a clear specification for the Smart Urban Mobility MVP. Section 1 (Introduction) outlines the project context, objectives, and terminology. Section 2 (General Description) details the system's environment, functions, user characteristics, constraints, and assumptions. Sections 3–5 specify requirements for ITSS, SPMS, and SPS, respectively, covering functional, performance, and security aspects. Section 6 the plan employs a hybrid methodology combining Kanban for task management, a condensed Design Sprint for ideation and prototyping. Section 7 presents use case scenarios for typical user interactions. The Annexes include a glossary, data dictionary, and supplementary references.

2 Solution 1: Intelligent Traffic Signal System (ITSS)

2.1 General Objective

The Intelligent Traffic Signal System (ITSS) aims to enhance urban traffic efficiency and safety in Oran, Algeria, by leveraging advanced artificial intelligence (AI) algorithms and real-time data processing. The system optimizes traffic flow, improves emergency response times, ensures compliance with traffic regulations, and mitigates accident risks through the following functionalities:

- Prioritize emergency vehicles to expedite passage through intersections.
- Dynamically adjust traffic signal timings based on real-time traffic density.
- Detect and record traffic violations, such as red-light running, for enforcement.
- Identify accidents and issue automated alerts to emergency services and drivers.

2.2 Detailed Specifications

2.2.1 Dynamic Light Timers Based on Traffic Density

- **Density Measurement**: High-definition cameras process 30 FPS feeds using the YOLOv8 model to detect and classify vehicles (cars, trucks, buses, motorbikes) with 95% accuracy. The system counts vehicles (n_{cars}) in real time to estimate queue lengths at intersections.

- **AI-Driven Adjustment**: A Q-learning algorithm optimizes green light duration (T_g) based on the formula:

  $$T_g = T_b + k \cdot (n_{cars} - n_{avg})$$

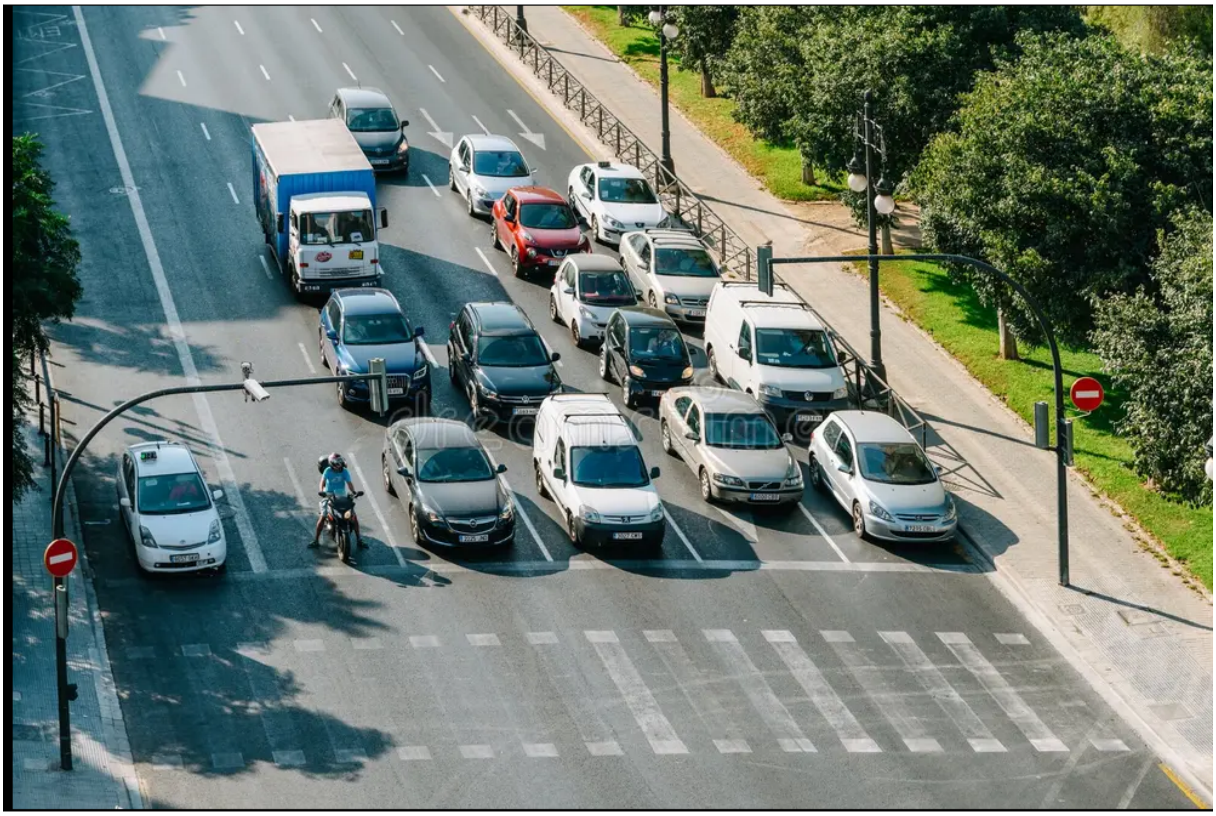
  where:

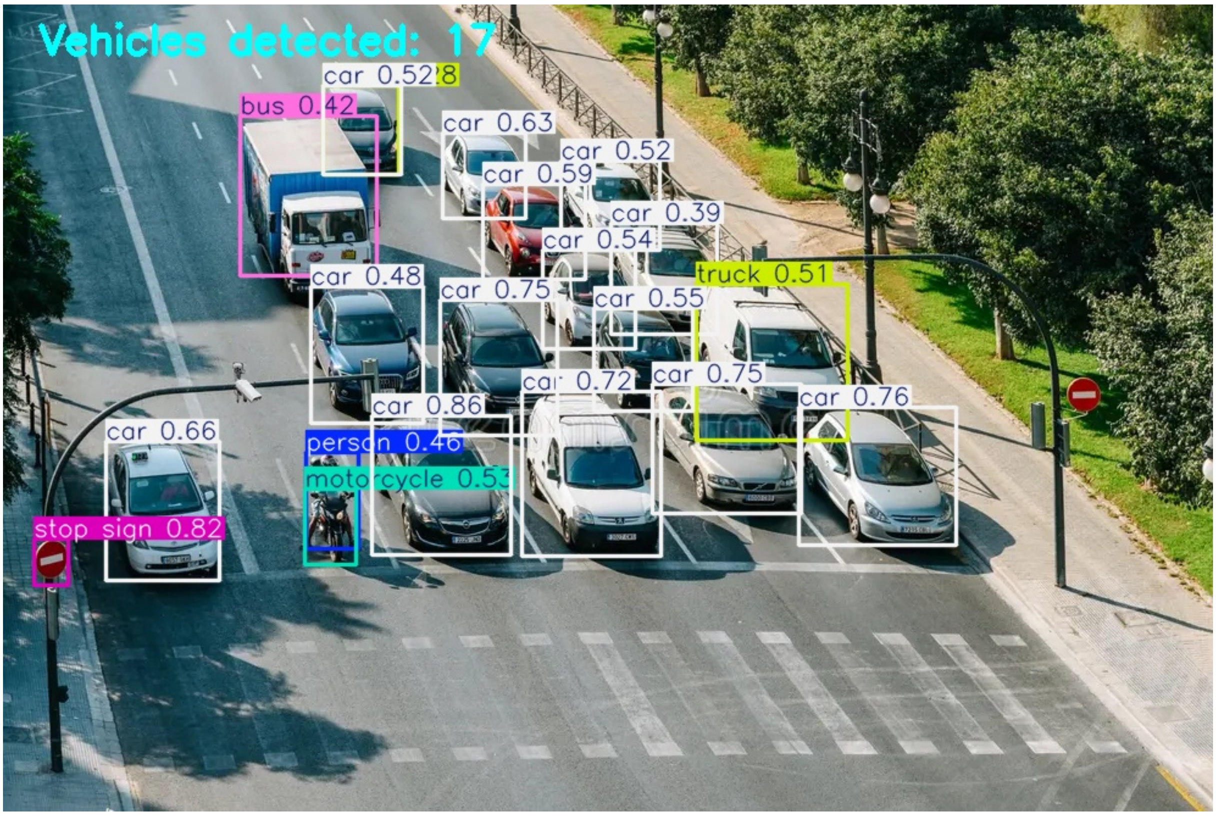
  - T_b: Base green time (e.g., 20 seconds).
  - k: Adjustment coefficient (e.g., 2 seconds per vehicle).
  - n_{cars}: Current vehicle count.
  - n_{avg}: Threshold for average queue length (e.g., 9 vehicles).

  For heavy traffic ($n_{cars} \geq 2 \cdot n_{avg}$), T_g is capped at 70% of the calculated value to prevent excessive delays on opposing directions. This approach reduces average wait times by up to 20%.

- **Adaptive Cycle Lengths**: Signal cycles adjust between 60–120 seconds based on peak or off-peak conditions, informed by historical traffic data.

- **Performance Metric**: Achieves 95% accuracy in vehicle counting and queue estimation, validated against ground-truth datasets.





2.2.2 Emergency Vehicle Prioritization

- **Detection Mechanism**: High-definition cameras (1080p, 30 FPS) employ Convolutional Neural Networks (CNNs), specifically Faster R-CNN, to detect emergency vehicles (e.g., ambulances, fire trucks, police cars) by analyzing visual features (e.g., flashing lights, sirens). Optical Character Recognition (OCR) identifies license plates to confirm vehicle registration, achieving 95% detection accuracy within 100 meters.
- **Signal Override**: Upon detection, ITSS sets the traffic signal to green in the emergency vehicle's direction, holding opposing directions red to ensure safe passage.
- **Duration Control**: Green light duration is adjusted (10–30 seconds) based on GPS-tracked speed and proximity, using real-time trajectory data.
- **Reversion to Normal Operation**: The system reverts to AI-driven signal timing within 5 seconds after the vehicle clears the intersection.
- **Data Logging**: Events are stored in JSON format for audit purposes:


```
{"timestamp": "2025-04-25T10:00:00Z", "vehicle_type": "ambulance",
 "intersection_id": "INT123"}
```


2.2.3 Real-time Violation Detection

- **Violation Identification**: YOLOv8-based video analytics track vehicle trajectories to detect violations (e.g., red-light running, illegal turns) with 90% precision. OCR extracts license plate numbers with 95% readability under optimal lighting.
- **Evidence Capture**: The system records 5-second 1080p video clips and still images, timestamped for legal documentation.

- **Alert Generation**: Violation data is stored in a MongoDB database:

```
{"timestamp": "2025-04-25T10:00:00Z", "location": "INT123",
 "plate": "ABC123", "violation_type": "red_light"}
```

  Alerts are transmitted to the nearest police department via a secure API.

- **Privacy Compliance**: Non-violator data is anonymized in accordance with Algerian data protection laws (Loi n^o 18-07).

2.2.4 Accident Detection and Automated Alerts

- **Anomaly Detection**: AI algorithms, leveraging YOLOv8 and motion analysis, identify accidents (e.g., collisions, sudden stops) with 90% accuracy by detecting irregular vehicle behavior in camera feeds.
- **Verification**: Future enhancements may integrate acoustic sensors for noise-based confirmation (not included in MVP).
- **Alert and Signage Activation**: Upon detection, ITSS sends alerts to a central monitoring system and activates LED signage (e.g., Panneau Triflash) to warn approaching drivers. Alerts include GPS coordinates and images.
- **Response Time**: Alerts are generated within 5 seconds, ensuring rapid response by emergency services.

2.3 Implementation Details

The ITSS leverages a Python-based AI pipeline for vehicle detection and signal timing optimization, implemented using the YOLOv8 model and OpenCV. The core algorithm processes camera feeds to detect vehicles, compute dynamic green light durations, and visualize results for operator monitoring. Key components include:

- **Vehicle Detection**: The YOLOv8 model (‘yolov8n.pt‘) classifies vehicles (cars, trucks, buses, motorbikes) with high accuracy, processing 1080p images in real time.
- **Green Light Timing**: The ‘calculate$_t g$‘$functionimplementsthedynamictimingformula, adjustingT_g$ based on vehicle counts and traffic thresholds.
- **Visualization**: Annotated images display bounding boxes and vehicle counts, aiding traffic controllers in monitoring system performance.

The following code snippet outlines the core pipeline:

```
import cv2
import numpy as np
from ultralytics import YOLO

def count_vehicles(image, model):
    results = model(image)[0]
    vehicle_classes = ['car', 'truck', 'bus', 'motorbike']
    n_cars = sum(1 for c in results.boxes.cls if model.names[int(c)] in vehicle_classes)
    return n_cars, results

def calculate_tg(n_cars, Tb, k, n_avr):
```

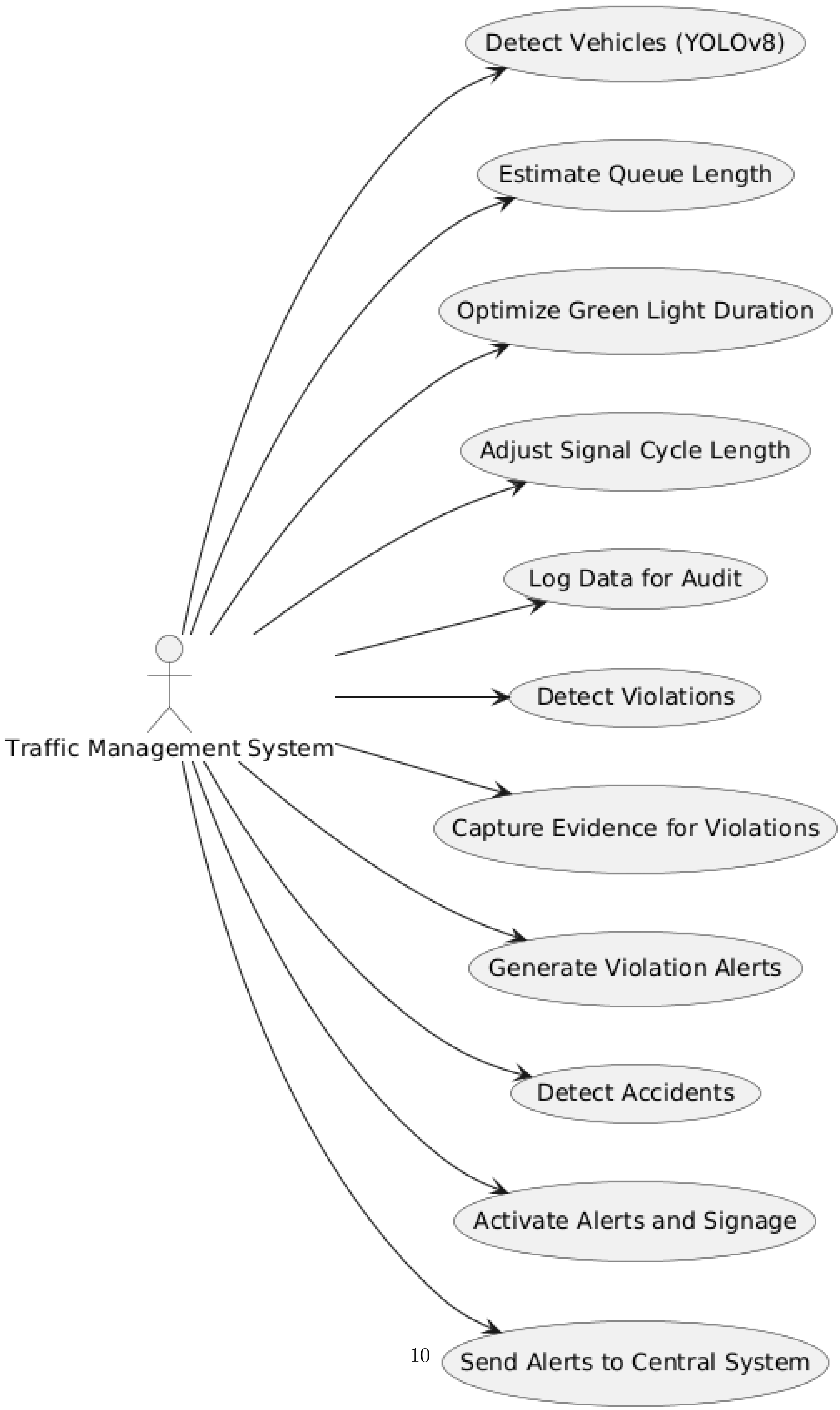
```
    base_time = Tb + k * (n_cars - n_avr)
    if n_cars >= 2 * n_avr:
        Tg = 0.7 * base_time
    else:
        Tg = base_time
    return {'Tg': round(Tg, 2), 'Tb': Tb, 'k': k, 'n_cars': n_cars, 'n_avr': n_avr}

model = YOLO('yolov8n.pt')
image = cv2.imread('traffic.jpg')
n_cars, results = count_vehicles(image, model)
output = calculate_tg(n_cars, Tb=20, k=2, n_avr=9)
```

2.4 Future Plans

To ensure the long-term scalability and effectiveness of ITSS within Oran's smart city framework, the following enhancements are planned for future development:

- **Vehicle-to-Infrastructure (V2I) Communication**: Integrate dedicated short-range communications (DSRC) or 5G-based V2I protocols to allow emergency vehicles to directly request signal prioritization, reducing detection latency and improving response times by 10%.
- **Multi-Intersection Coordination**: Extend the Q-learning algorithm to optimize signal timings across multiple intersections, using a centralized traffic management system to minimize city-wide congestion, targeting a 30% reduction in average travel times.
- **Weather-Based Adjustments**: Incorporate real-time weather data (e.g., rain, fog) into the dynamic timing algorithm, adjusting T_g to account for reduced visibility or slippery roads, enhancing safety during adverse conditions.
- **Automated Ticketing System**: Develop an automated ticketing module for traffic violations, integrating with Oran's police database to issue fines based on OCR-captured license plates, streamlining enforcement processes.
- **Enhanced Accident Detection**: Implement acoustic sensors and vehicle telemetry to improve accident detection accuracy to 95% by combining visual, auditory, and motion data.
- **Multilingual Interfaces**: Support Arabic and French interfaces for the traffic controller dashboard and commuter notifications, addressing Oran's linguistic diversity.
- **National Database Integration**: Connect ITSS with Algeria's national transportation database to enable cross-city violation tracking and traffic pattern analysis, facilitating coordinated urban planning.
- **Advanced AI Models**: Upgrade to YOLOv9 or other state-of-the-art models for vehicle detection and explore deep reinforcement learning for signal optimization, aiming for 98% detection accuracy and 25% wait time reduction.


Traffic Management System
Detect Vehicles (YOLOv8)
Estimate Queue Length
Optimize Green Light Duration
Adjust Signal Cycle Length
Log Data for Audit
Detect Violations
Capture Evidence for Violations
Generate Violation Alerts
Detect Accidents
Activate Alerts and Signage
Send Alerts to Central System

2 Solution 2: Smart Predictive Maintenance System (SPMS)

2.1 General Objective

The Smart Predictive Maintenance System (SPMS) is an intelligent dashboard platform designed to enhance the operational reliability of the Oran tramway network. Using AI-powered predictive analytics and real-time monitoring, SPMS identifies infrastructure components—specifically the segments between tram stations—that are likely to require maintenance. By forecasting failures before they occur, the system improves service continuity, reduces unplanned downtime, and ensures passenger safety through the following functionalities:

- Predict degradation between tram segments using neural network models trained on infrastructure sensor data.
- Visualize segment condition across the tram map using color-coded urgency levels.
- Provide real-time updates through an interactive dashboard, supporting proactive maintenance planning.

2.2 Detailed Specifications

2.2.1 AI-Based Maintenance Classification

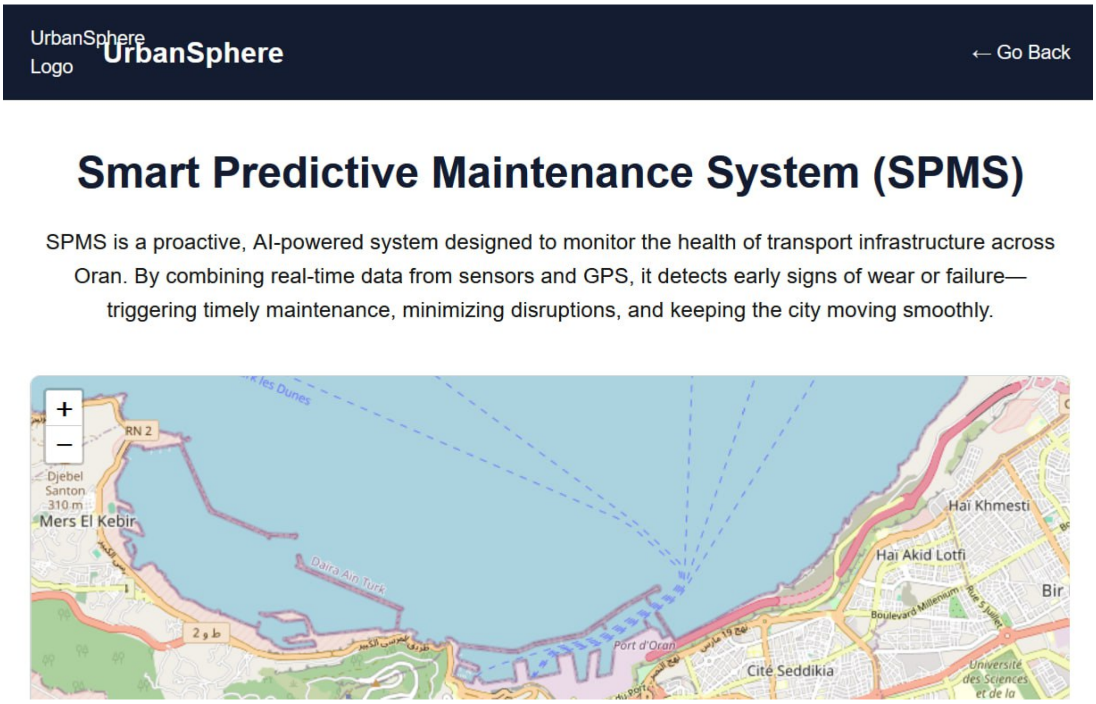
- **Preprocessing**:
  - Data is split using `train_test_split()` with stratified sampling.
  - Features normalized using `StandardScaler()`.
- **Model Architecture**: A PyTorch neural network classifier, achieving **97% accuracy** across three priority classes.
- **Output Classes**:
  - 0: High Priority – Maintenance needed before 2025 (Red).
  - 1: Medium Priority – Maintenance needed before 2027 (Orange).
  - 2: Low Priority – Maintenance needed before 2030 (Green).
- **Output Format (sample)**:

```
{
  "timestamp": "2025-04-25T10:00:00Z",
  "segment": ["ST045", "ST046"],
  "priority": 0
}
```

2.2.2 Visualization Dashboard

- **Map Interface**: An interactive dashboard displays the tram line with segments colored as follows:
  - Red: Priority 0 – Needs maintenance before 2025
  - Orange: Priority 1 – Before 2027
  - Green: Priority 2 – Before 2030
  - Gray: Out of service
- **Segment Inspection**: Displays station pair, predicted priority, and prediction timestamp.

- **Prediction Frequency**: Model runs weekly for continuous updates.



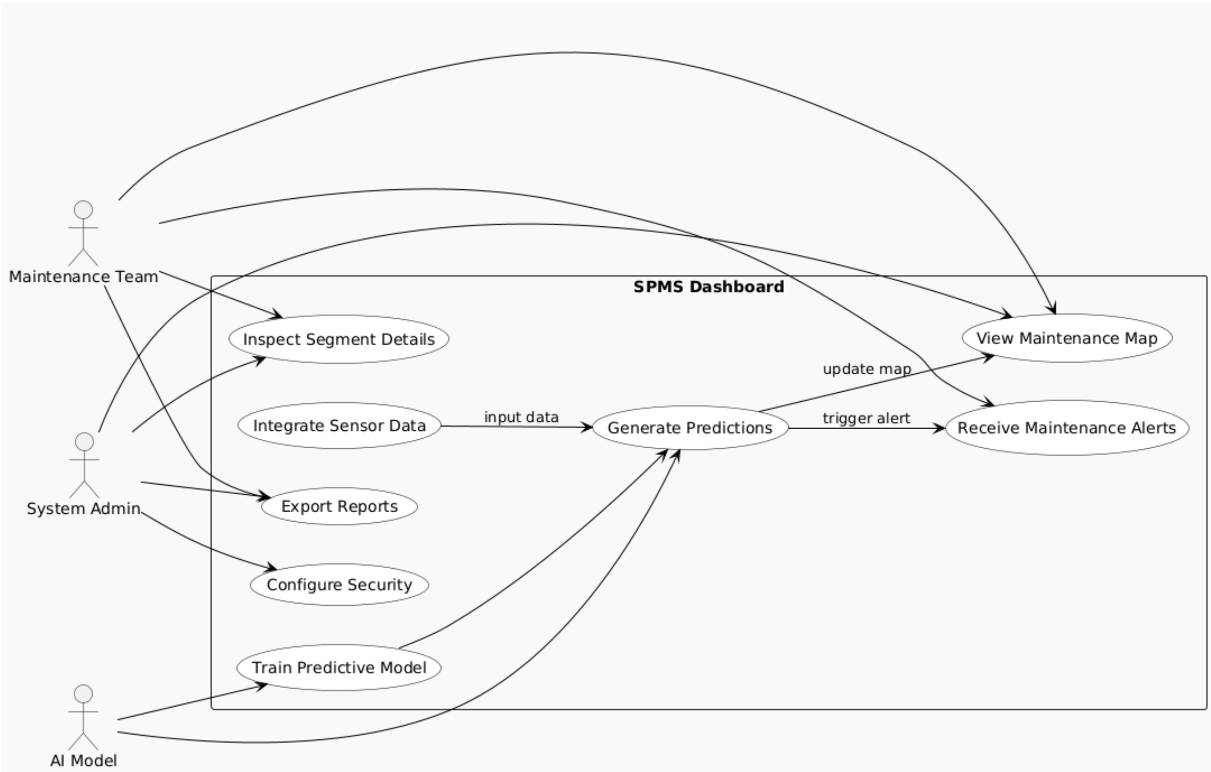
2.3 Implementation Details

- **Frameworks Used**: PyTorch, Pandas, Scikit-learn, Leaflet.js (frontend map), Flask (backend API).
- **Security Note**: TLS 1.3 and JWT authentication will be added in future iterations.

2.4 Future Plans

The SPMS roadmap includes several strategic enhancements:

- **Alert System Integration**: Automatic alerts when a segment escalates to a higher risk category.
- **Security & Authentication**: Implement encrypted TLS 1.3 channels and JWT-based authentication.
- **Sensor Stream Integration**: Real-time IoT sensor data ingestion.
- **Failure Probability Estimation**: Predict not only priority, but likelihood of failure.
- **Multilingual Dashboard**: Support for Arabic and French UI elements.
- **ITSS Integration**: Link with Oran's traffic signal system to adjust flow during maintenance periods.
- **Mobile App**: Interface for field technicians.
- **3D Viewer**: Visual analysis of structural wear in critical segments.



3 Solution 3: Urban Mobility Companion App (UMCA)

3.1 General Objective

The Urban Mobility Companion App (UMCA) is designed to streamline transportation within the city of Oran by offering an intuitive and intelligent mobile application for both public transport passengers and professional drivers. Developed using Flutter for cross-platform support, the app uses real-time geolocation, dynamic mapping, and city data services to provide route planning, transit insights, and parking assistance. UMCA enhances the commuting experience through a personalized and responsive interface that addresses the distinct needs of each user type.

3.2 Detailed Specifications

3.2.1 Passenger Journey Planner

- **Input Interface:** Passengers input their desired destination directly within the app.
- **Geolocation Service:** The application captures the user's current location using device GPS.
- **Transit Suggestions:** Based on the passenger's position and input, the app displays available lines (bus or tram) connecting to the destination.
- **Line Visualization:** Upon selection, the app presents:
  - The current line and station route
  - Real-time vehicle location
  - Progress along the selected line
- **Arrival Confirmation:** When the user arrives at the desired station, a success message is displayed.

- **Accuracy:** Location data is refined for high precision with a typical accuracy of under 5 meters.

3.2.2 Driver Parking Assistant

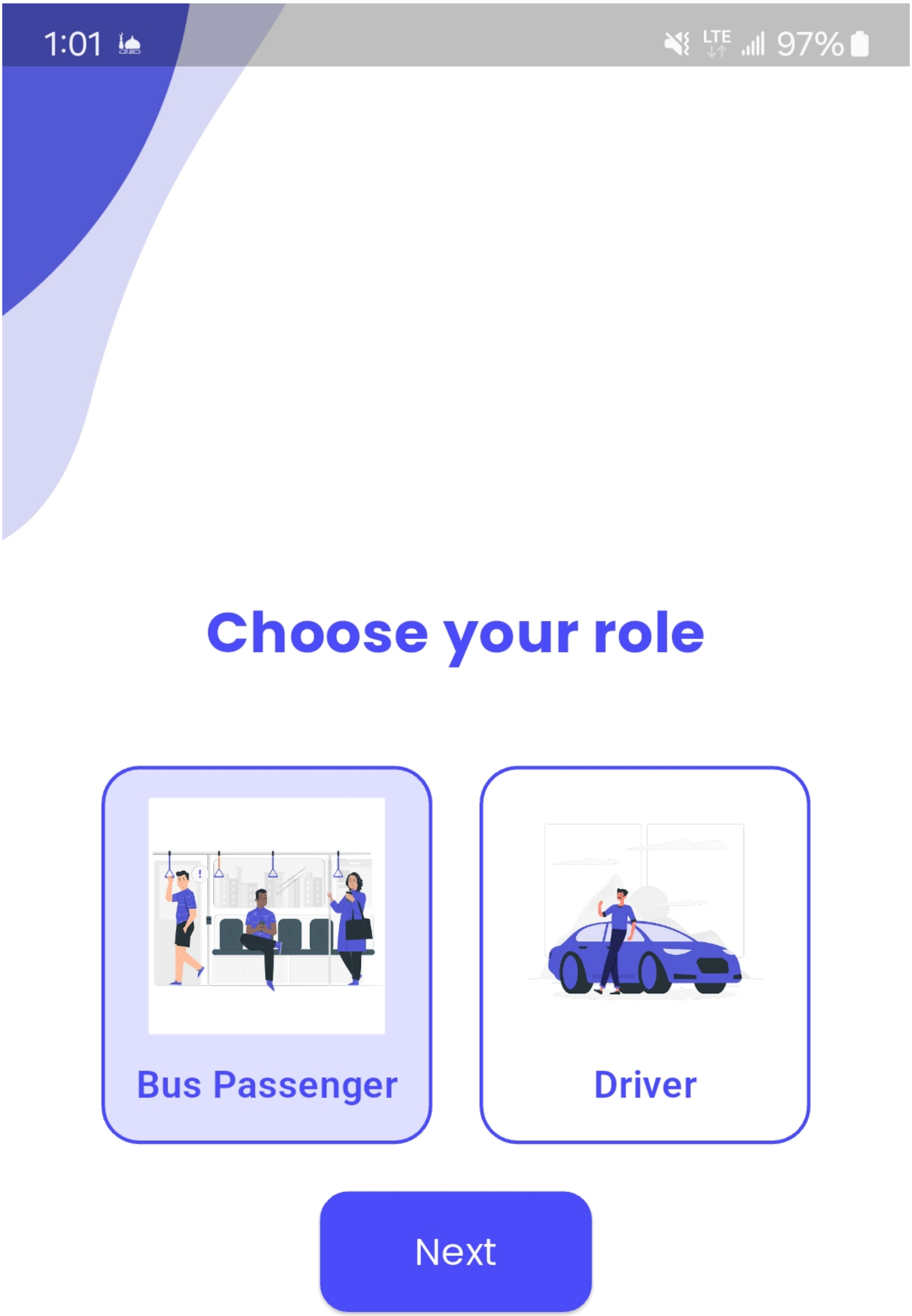
- **Destination Input:** Drivers enter their intended destination through a dedicated input field.
- **Available Parking Display:** The system retrieves and shows real-time data of available parking spots near the target location.
- **Map Guidance:** A dynamic map highlights:
  - Suggested parking locations
  - Navigation route to the destination
  - Estimated arrival time
  - Parking type (public, private, reserved, accessible)
- **Dynamic Updates:** Routes adapt in real time based on traffic conditions and parking availability.

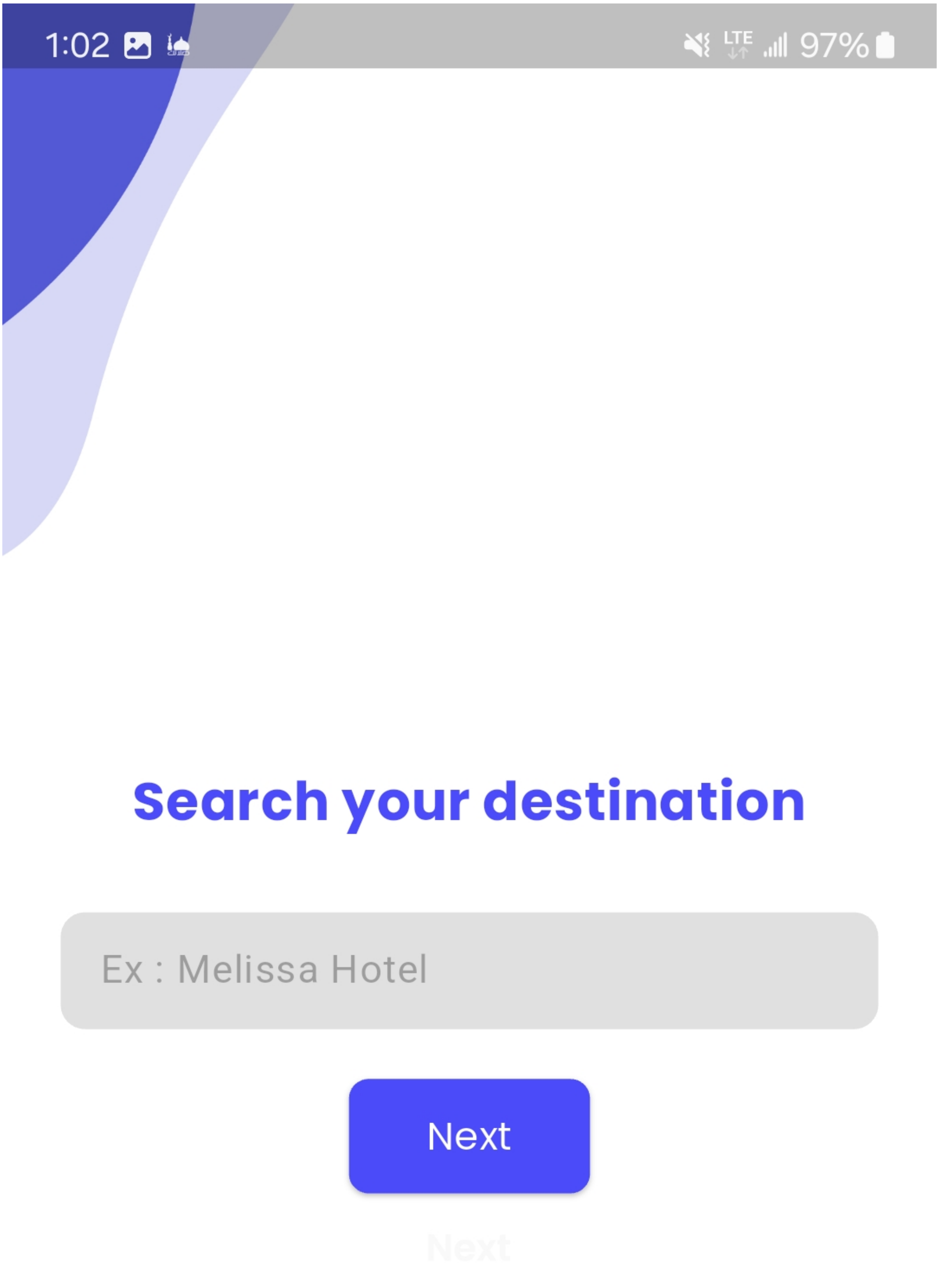
3.3 Implementation Details

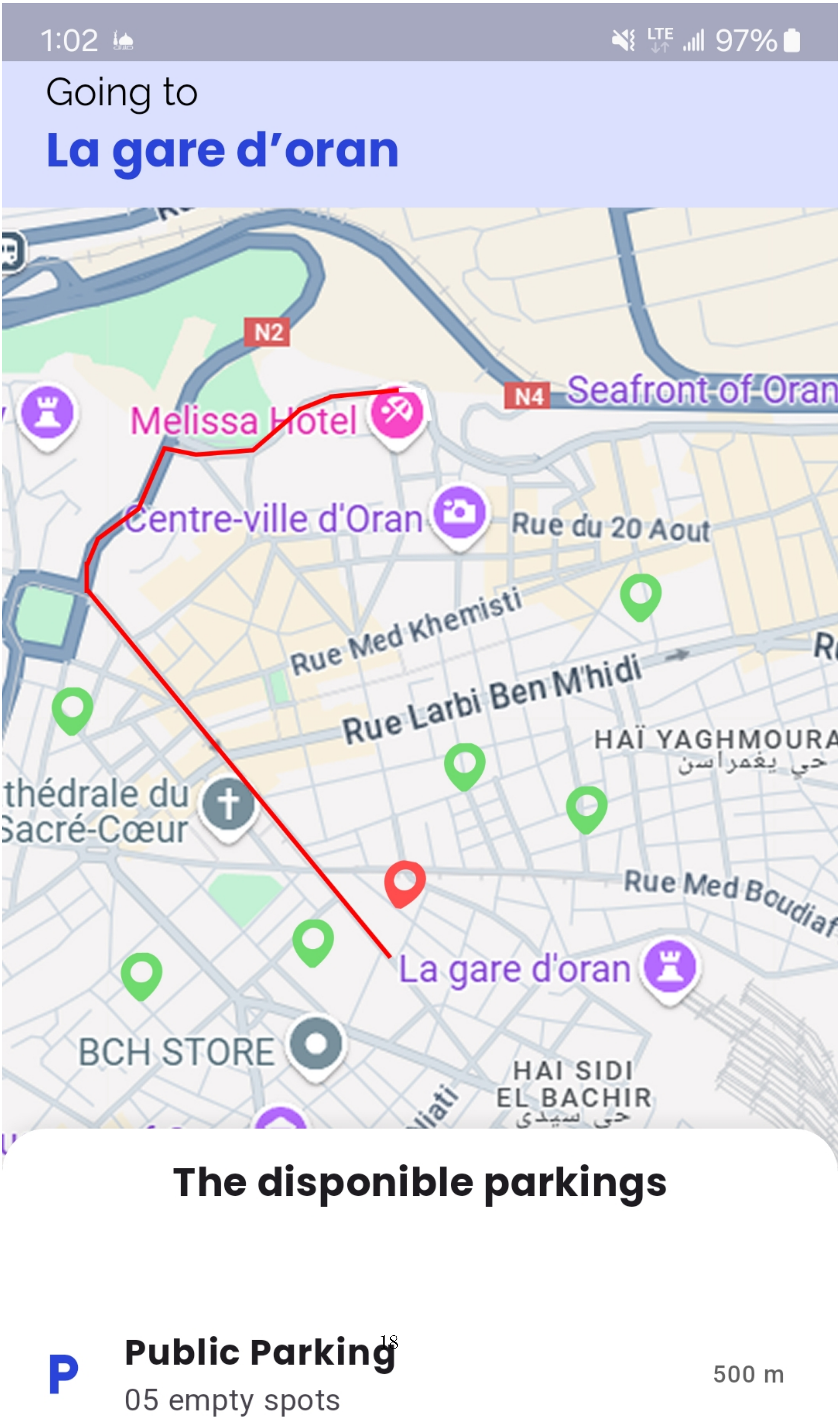
UMCA is developed using the Flutter framework to ensure seamless functionality across Android and iOS devices. Real-time data synchronization and navigation features are powered by cloud-based services and location APIs. Key components include:

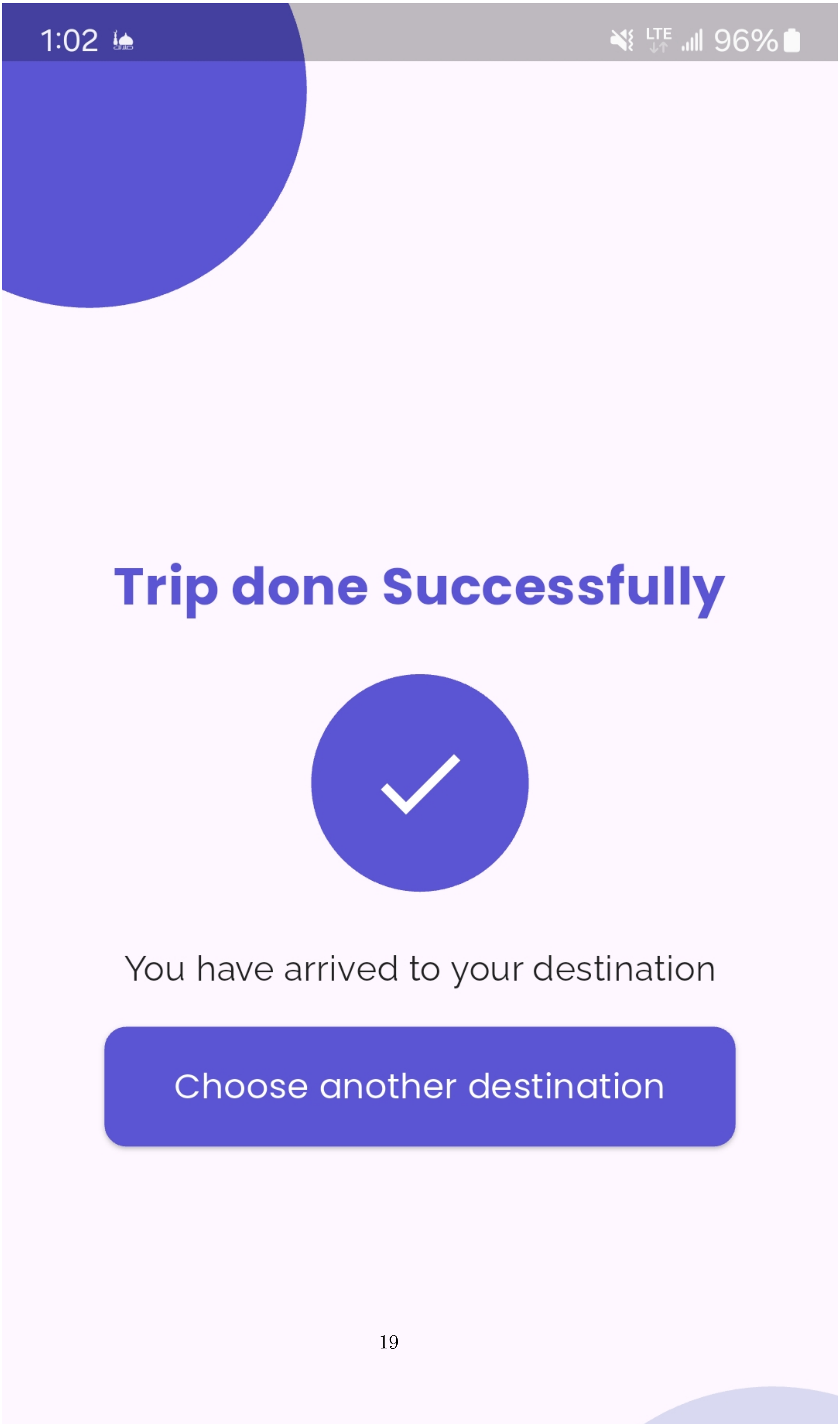
- **Framework:** Flutter (Dart)
- **Map and Navigation:** Google Maps SDK
- **Real-Time Services:** Firebase Firestore and Geolocator for user tracking and updates
- **User Interface:** A dual-tab layout allows users to switch between Passenger and Driver modes.
- **Cloud Backend:** Firebase for authentication and data storage.

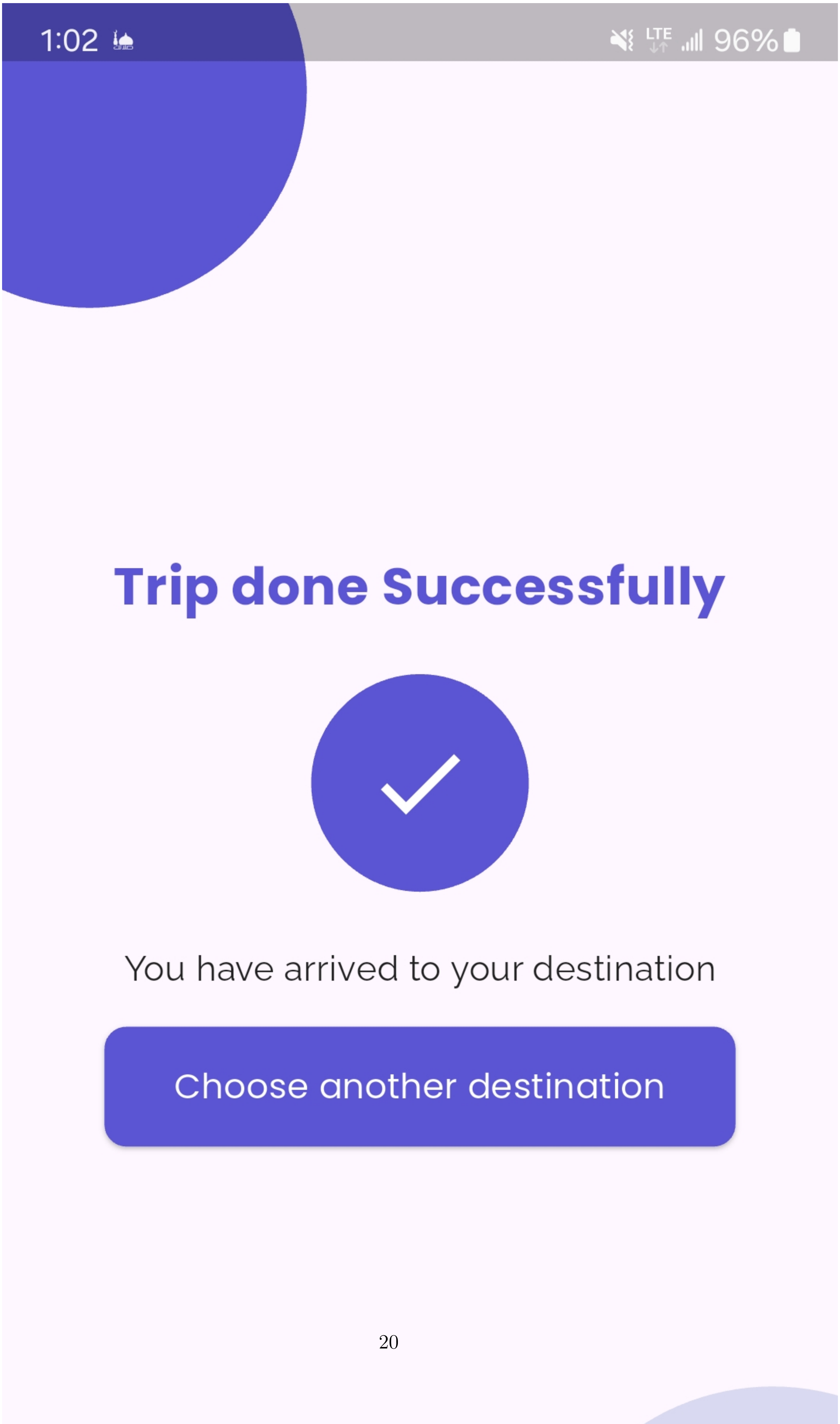
3.4 Application Showcases

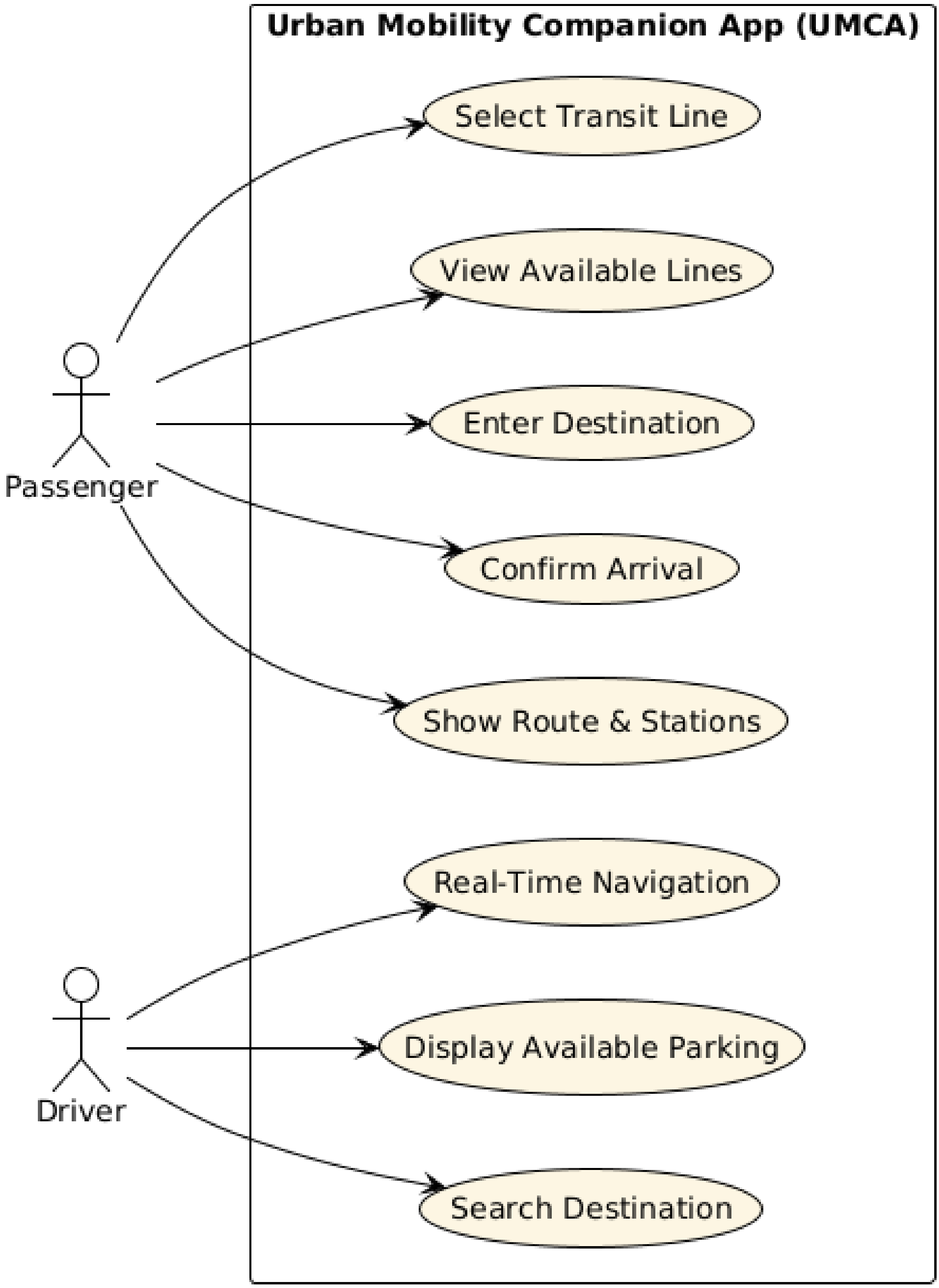



1:02
97%
Search your destination
Ex : Melissa Hotel
Next


1:02
97%
Going to
La gare d'oran
N2
N4
Seafront of Oran
Melissa Hotel
Centre-ville d'Oran
Rue du 20 Aout
Rue Med Khemisti
Rue Larbi Ben M'hidi
HAÏ YAGHMOURA
حي يغمراسن
thédrale du
Sacré-Cœur
Rue Med Boudiaf
La gare d'oran
BCH STORE
HAI SIDI
EL BACHIR
The disponible parkings
P
Public Parking
05 empty spots
500 m


1:02
96%
Trip done Successfully
You have arrived to your destination
Choose another destination


1:02
96%
Trip done Successfully
You have arrived to your destination
Choose another destination



3.5 Future Plans

UMCA will continue to evolve with new capabilities designed to better support the growing needs of urban mobility in Oran. Planned future developments include:

- **Multimodal Integration:** Incorporation of taxis, e-scooters, and tram systems to offer comprehensive route planning.

- **Offline Access:** Enable users to download routes and station data for offline use.
- **Voice Assistant:** Add a voice-guided assistant for hands-free commands and audio station alerts.
- **Crowd Detection:** AI-powered prediction of passenger density per vehicle to reduce overcrowding.
- **Driver Alerts:** Push notifications to drivers for road closures and dynamic parking updates.
- **Multilingual Support:** Interface available in both Arabic and French to serve a diverse user base.
- **Emergency SOS Button:** Direct line for emergency services with live location sharing.
- **Traffic System Integration:** Future linkage with the ITSS to provide smart synchronization for public transport priority at intersections.

4 Use Case Scenarios

This section presents use case scenarios for the Smart Urban Mobility Minimum Viable Product (MVP), illustrating typical interactions with the Intelligent Traffic Signal System (ITSS), Smart Predictive Maintenance System (SPMS), and Smart Parking and Transit System (SPS). Each use case describes the actors, preconditions, main flow, postconditions, and exceptions, providing a clear understanding of how the system addresses Oran's transportation challenges during the INNOVERSE Hackathon 2025. The scenarios align with the functional specifications and IEEE 830-1998 standards, ensuring clarity for developers, testers, and stakeholders.