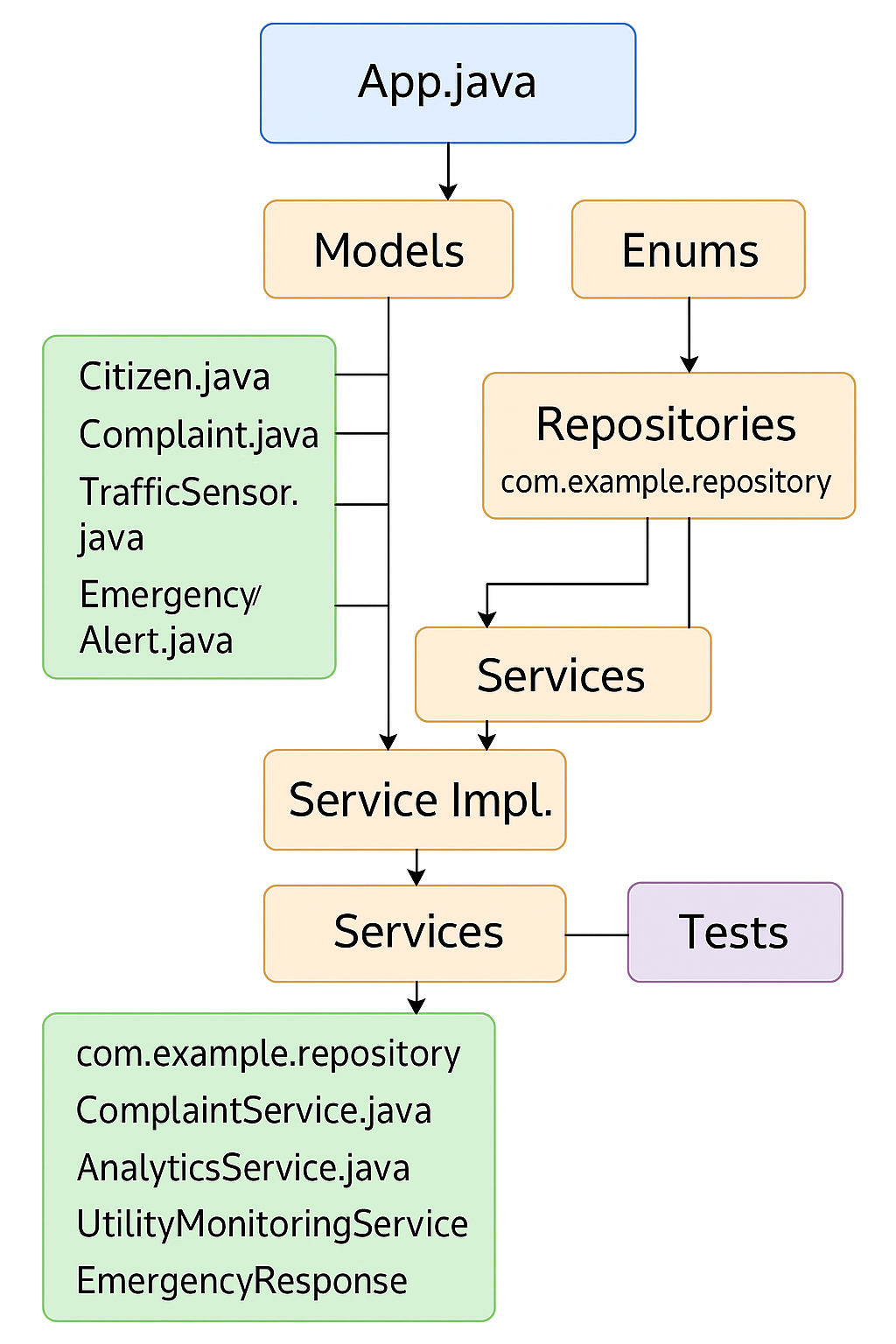
# SCMS CLI Backend – Smart City Management System

This is a modular, test-driven backend project for a Smart City Management System (SCMS) built using Java. It follows a clean CLI-based architecture using object-oriented principles, TestNG for validation, and in-memory repositories.

## 🧱 Layers Overview

- Entry Point: `App.java` serves as the CLI application driver.  
- Models: Core domain objects such as Citizen, Complaint, EmergencyAlert, etc.  
- Enums: Categorization for statuses, types, and priorities.  
- Repository Layer: Manages data with in-memory implementations.  
- Services Layer: Contains business logic for complaints, traffic, utilities, and emergencies.  
- Implementation Layer: Specific logic implementation for each service.  
- Test Layer: Validates service logic using TestNG.  
- Exception Handling: Graceful error handling through custom exceptions.

## 📐 Architecture Diagram



## 📁 Folder Structure

SCMS-CLI-Backend/  
├── src/  
│ └── test/java/com/city/  
│ ├── model/ # Citizen, Complaint, Emergency, TrafficSensor  
│ ├── model/enums/ # AlertStatus, EmergencyType, etc.  
│ ├── repository/ # Interfaces for CRUD logic  
│ ├── repository/memory/ # InMemory\*Repository classes  
│ ├── service/ # Interfaces & Test classes  
│ ├── service/impl/ # Implementation logic  
│ ├── exception/ # Custom exceptions  
│ └── App.java # CLI Entry point  
├── testng.xml  
├── pom.xml  
├── README.md

## ✅ Test Coverage

- `AnalyticsServiceTest.java` – validates statistical logic  
- `TrafficManagementServiceTest.java` – tests congestion detection  
- `UtilityMonitoringServiceTest.java` – ensures threshold breach detection  
- `CitizenRepositoryTest.java` – verifies citizen registration and lookup

## 🚀 How to Run

1. Clone the project and open in IntelliJ IDEA.  
2. Run `App.java` to launch the CLI interface.  
3. Use TestNG to run unit tests via `testng.xml`.

## 🧪 Technologies Used

- Java (OOP)  
- TestNG (Testing Framework)  
- Maven (Build Management)  
- IntelliJ IDEA (IDE)

## 👤 Author

Created by Akash Kuna – Focused on clean, test-driven Java applications.