**IoT based Traffic management**

* Our goal is to improve car traffic in Iași, collecting real-time data shared by any participant and making it available for anyone using the app. (1)

**Actors:**

* **Normal Car:** traffic participants that drive cars must always respect the rules (such as passing only on green light, giving priority when there is a sign imposing this); (2)

*drive* → drive on the streets (3)

*respectTrafficIndicator* → the driver must pay attention to indicators and respect them

* **Ambulance:** traffic participants that can drive with increased speed when an emergency occurs and people’s lives are at risk;

*drive* (without respecting traffic limitations)

*turnOnSounds* → turns on the audio signals

*turnOnLights* → turns on the rotating beacons

* **Fire Truck:** traffic participants that can also drive with increased speed when a fire starts;

*drive* (without respecting traffic limitations)

*turnOnSounds* → turns on the audio signals

*turnOnLights*  → turns on the rotating beacons

* **Police car:**  traffic participants that can also drive with increased speed when an emergency occurs and people’s lives are at risk;

*drive* (without respecting traffic limitations)

*turnOnSounds* → turns on the audio signals

*turnOnLights*  → turns on the rotating beacons

* **Traffic sign:**  signs can indicate priority giving/having, the obligation to stop, information about distance to a nearby city;

*giveInstructionsBySigns* → controls traffic using signs

* **Traffic light:** green/yellow/red, where green means go, yellow means slow down and red means don’t pass;

*giveInstructionsByColors* → controls traffic using the three colors

* **Google Maps:** export data about car accidents and collect real-time information about traffic events (accidents, road blocks);

*exportData/importData* → export/import data

* **Cameras:** checks how crowded an area is so the system can adjust;

*monitorTraffic* → tracks the number of cars and their speed

*sendData* → send video signal to the system

* **Main system:** the main component which saves and processes the data from the cameras to adjust the time for the traffic light and find alternative routes for cars;

*importData/exportData* → receives/sends data from/to traffic lights and digital indicators

*processData* → analyzes the data and decides the appropriate course of action

*saveData* → caches videos when something happens and keeps some data approximating rush hours

**Server:**

* **Main system:** the main component which saves and processes the data from the cameras to adjust the time for the traffic light and find alternative routes for cars;

*ireceiveVideoData* → receives data from cameras

*sendMapData* → sends data about maps to clients

*sendColorSignal* → sends traffic lights signals to change colors

* **Google Maps:** export data about car accidents and collect real-time information about traffic events (accidents, road blocks);

*exportMapData* → sends data about maps to main system

**Client:**

**Hardware client:**

* **Traffic sign:**  signs can indicate priority giving/having, the obligation to stop, information about distance to a nearby city;

*giveInstructionsBySigns* → controls traffic using signs

* + **Traffic light:** green/yellow/red, where green means go, yellow means slow down and red means don’t pass;

*giveInstructionsByColors* → displays color for car client

*receiveSignal* → receives signal to change color from the server

* + **Cameras:** checks how crowded an area is so the system can adjust;

*sendVideoData* → send video signal to the system

**Car client:**

* **Normal Car:** traffic participants that drive cars must always respect the rules (such as passing only on green light, giving priority when there is a sign imposing this); (2)

*exportLocation* → sends the location to the server

*importMap* → receives data about the map from server

*importRecommendation* → receives recommended route from server

* **Ambulance:** traffic participants that can drive with increased speed when an emergency occurs and people’s lives are at risk;

*drive* (without respecting traffic limitations)

*exportLocation* → sends the location to the server

*importMap* → receives data about the map from server

*importRecommendation* → receives recommended route from server

*getEmergencyLocation* → receives from the server a location where an emergency situation is happening

* + **Fire Truck:** traffic participants that can also drive with increased speed when a fire starts;

*exportLocation* → sends the location to the server

*importMap* → receives data about the map from server

*importRecommendation* → receives recommended route from server

*getEmergencyLocation* → receives from the server a location where an emergency situation is happening

* **Police car:**  traffic participants that can also drive with increased speed when an emergency occurs and people’s lives are at risk;

*drive* (without respecting traffic limitations)

*exportLocation* → sends the location to the server

*importMap* → receives data about the map from server

*importRecommendation* → receives recommended route from server

*getEmergencyLocation* → receives from the server a location where an emergency situation is happening