

System Design Document: Intelligent Traffic Management System (ITMS)

***Students, Inc.***

***Last revised date: 2023-12-05***

**Authors:**

* Wouter Swinkels
* Iskren Zhechev
* Vladislav Stefanov
* Adrián Pacera
* Tim van Kol

Table of Contents

[List of tables 4](#_Toc151543013)

[List of figures 4](#_Toc151543014)

[Introduction 6](#_Toc151543015)

[Problem Statement: Challenges in Contemporary Traffic Management Control Systems 6](#_Toc151543016)

[Efficiency Constraints 6](#_Toc151543017)

[Data Inconsistency and Lack of Integration 6](#_Toc151543018)

[Economic and Environmental Impact 7](#_Toc151543019)

[Security Vulnerabilities 7](#_Toc151543020)

[Conclusion: The Imperative for the Intelligent Traffic Management System (ITMS) 8](#_Toc151543021)

[System context 9](#_Toc151543022)

[Fig 1. Context diagram illustrates the dynamic interactions between the system and stakeholders (i.e., drivers, pedestrians, and traffic authorities). 9](#_Toc151543023)

[System design 11](#_Toc151543024)

[System Use Cases 11](#_Toc151543025)

[Introduction 11](#_Toc151543026)

[Use case environment 1: simple vehicular-only 4-way intersection, no turning 12](#_Toc151543027)

[Introduction 12](#_Toc151543028)

[Scenarios 13](#_Toc151543029)

[Requirements 16](#_Toc151543030)

[User requirements 16](#_Toc151543031)

[Functional requirements 16](#_Toc151543032)

[Nonfunctional requirements 17](#_Toc151543033)

[Moscow analysis 18](#_Toc151543034)

[Sequence diagrams user case one 19](#_Toc151543035)

[Scenario 1 20](#_Toc151543036)

[Scenario 2 22](#_Toc151543037)

[Scenario 3 24](#_Toc151543038)

[Scenario 4 26](#_Toc151543039)

[Scenario 5 28](#_Toc151543040)

[Scenario 6 & 7 31](#_Toc151543041)

[Scenario 8 34](#_Toc151543042)

[Scenario 9 37](#_Toc151543043)

[Scenario 10 37](#_Toc151543044)

[System behavior 37](#_Toc151543045)

[System structure 39](#_Toc151543046)

[Traffic Direction Controller 40](#_Toc151543047)

[Managing an incoming traffic direction 40](#_Toc151543048)

[Decentralized Operations 40](#_Toc151543049)

[System Structure 41](#_Toc151543050)

[Component Diagram 41](#_Toc151543051)

[Message protocol 41](#_Toc151543052)

[Design rationale 43](#_Toc151543053)

[Component analysis 43](#_Toc151543054)

# List of tables

[Table 1, Version history 5](#_Toc151543061)

[Table 2, Terms and Abbreviations 5](#_Toc151543062)

[Table 3. Scenario 1 13](#_Toc151543063)

[Table 4. Scenario 2 13](#_Toc151543064)

[Table 5. Scenario 3 13](#_Toc151543065)

[Table 6. Scenario 4 14](#_Toc151543066)

[Table 7. Scenario 5 14](#_Toc151543067)

[Table 8. Scenario 6 14](#_Toc151543068)

[Table 9. Scenario 7 15](#_Toc151543069)

[Table 10. Scenario 8 15](#_Toc151543070)

[Table 11. Scenario 9 15](#_Toc151543071)

[Table 12. Scenario 10 15](#_Toc151543072)

[Table 13. User Requirements 16](#_Toc151543073)

[Table 14. Functional requirements 16](#_Toc151543074)

[Table 15. Nonfunctional requirements 17](#_Toc151543075)

# List of figures

[Figure 1. a Dutch ambulance responding to an emergency call having crashed on an intersection (DPG Media, 2023) 7](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546566)

[Figure 2. Context diagram that illustrates the dynamic interactions between the system and stakeholders (i.e., drivers, pedestrians, and traffic authorities). 9](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546567)

[Figure 3. Use case one 11](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546568)

[Figure 4. System behavior state machine abstract version 37](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546569)

[Figure 5. System behavior low level 38](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546570)

[Figure 6. Node structure 39](https://stichtingfontys.sharepoint.com/sites/IndustryProjectsem3autumn2023-7Students.INCCB/Gedeelde%20documenten/7%20Students.%20INC%20(CB)/System%20Analysis%20and%20Design/System%20Design%20Document%20-%20ITMS.docx#_Toc151546571)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Author(s) | Date | Description | Status |
| 0.1 | Tim, Iskren, Vladislav, Wouter, Adrián | 2023-10-10 | Creation and review | *Draft* |
| 0.2 | Tim, Iskren, Vladislav, Wouter, Adrián | 2023-10-31 | Addition of more diagrams and requirements, restructuring of UCs | *Draft* |
| 0.3 | Tim, Iskren, Vladislav, Wouter, Adrián | 2023-11-19 | Restructure of document, general simplification of requirements, MoSCoW | *Draft* |
| 0.4 | Vladislav | 2023-12-05 | Dividing of the former SDD into two separate documents: analysis and design document | *Draft* |

Table 1. Version history

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| SDD | System Design Document |
| AD | Analysis Document |
| PP | Project Plan |
| ITMS | Intelligent Traffic Management System |
| STS | Smart Traffic Signal |
| UC | Use case |
| S | Scenario |
| UR | User requirement |
| FR | Functional requirement |
| NFR | Non-functional requirement |
| SSD | State Sequence Diagram |
| SD | State Diagram |
| UCD | Use Case Diagram |

Table 2. Terms and Abbreviations

# Introduction

# System design

Table 16. MoSCoW analysis

Sequence diagrams user case one

In this section you will find sequence diagrams of the different use case scenarios.

### Scenario 1

A diagram of a computer program

Description automatically generated with medium confidence

A diagram of a data flow

Description automatically generated

### Scenario 2

A diagram of a software program

Description automatically generated with medium confidence

A white paper with black text

Description automatically generated

### Scenario 3

A screenshot of a computer program

Description automatically generated

A diagram of a program

Description automatically generated

### Scenario 4

A diagram of a program

Description automatically generated with medium confidence

A screenshot of a diagram

Description automatically generated

### Scenario 5

A diagram of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated

A diagram of a car

Description automatically generated

### Scenario 6 & 7

A screenshot of a computer program

Description automatically generated

A diagram of a data flow

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Scenario 8

A screenshot of a diagram

Description automatically generated

A diagram of a program

Description automatically generated

A diagram of a data flow

Description automatically generated

## Scenario 9

A screenshot of a computer

Description automatically generatedsyste

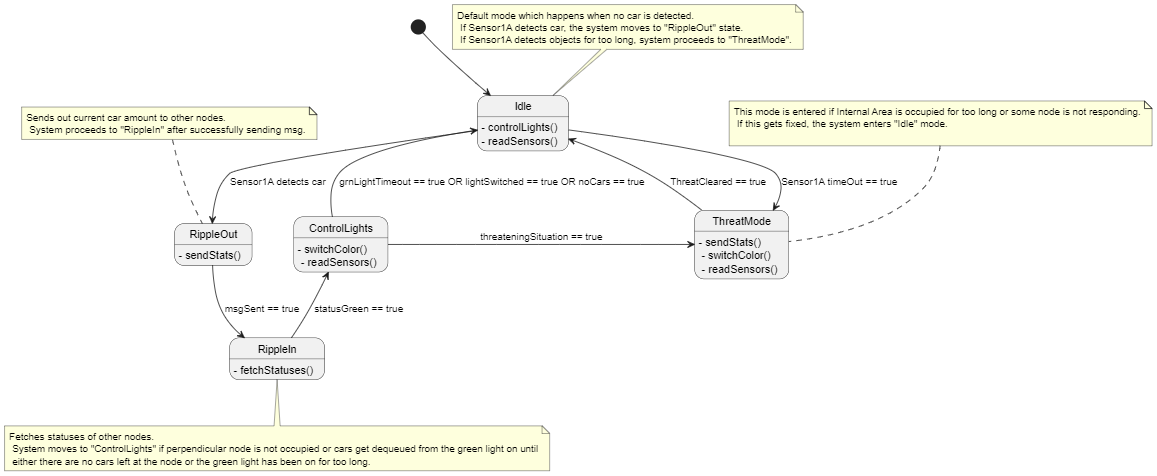
## Scenario 10

A screenshot of a diagram

Description automatically generated

# System behavior

In this section, a description of the system-state machine diagram can be found. For an explanation of each state, please see the comments in the diagram. Please note that this diagram is a rough sketch of what the system should do based on the functionalities described in use-case scenarios and user requirements. Due to the agile workflow, this diagram works more on the base of “start point”, from which the group will be able to develop a more in-depth diagram.



|  |
| --- |
|  |
|  |  |

Figure 4. System behavior state machine abstract version

Figure 5. System behavior low level

# System structure

The ITMS operates using a decentralized and interconnected network of nodes that all collaborate in real-time to ensure reliable and safe operations in the ITMS. Above, figure 1 describes an example implementation of a variety of nodes being part of the ITMS. The denoted implementation describes three types of nodes, being the Traffic Direction Controller (TDC), the Traffic Flow Controller (TFC) and the Traffic Gateway Management Controller (TGMC). Each of these nodes has a variety of local responsibilities, which are covered in the paragraphs below. Please note that this is not a comprehensive overview of all possible nodes, as the system could implement any node that would extend the functionality of the ITMS.

Afbeelding met diagram, schermopname, lijn

Automatisch gegenereerde beschrijving

Figure 6. Node structure

## Traffic Direction Controller

### Managing an incoming traffic direction

Each traffic light is equipped with various types of sensors to monitor vehicle and pedestrian traffic in its vicinity, which is usually. Every TDC uses the sensor data and a set of predefined algorithms to decide the optimal light sequence that the intersection should follow.

For instance, if a particular approach lane becomes devoid of vehicles, the TDC

### Decentralized Operations

A significant innovation is the decentralized mechanism of the ITMS that the TDC nodes will use.

Each TDC communicates with its neighboring TDC nodes that could extend the traffic junction consisting of the lights. This sharing of junction data allows multiple intersections to predict upcoming traffic conditions better together. For example, if a traffic light on junction A detects a surge of cars coming from one direction, it can inform a downstream junction B to prepare for increased flow.

In an intersection, Traffic Direction Controllers are responsible for guiding any incoming traffic appropriately. They

The Traffic Flow Controller (TFC) handles the interchange information between any intersections that want to either share data between each other or have data be sent further down the node system and is also responsible.

# System Structure

## Component Diagram

## Message protocol

The project uses a low-level message protocol as a more compact way of representing information. It utilizes messages consisting of two characters that describe the current state of the system or that an acknowledgement is received by another node. This protocol is simplified and allows the traffic management system to store and process data efficiently.

Message format: {{X}{Y}} i.e., {{Current node}{Next node}}

Messages are sent in three situations:

1. When a node upgrades the values of {X} and/or {Y} and shares the update to all other nodes to make them all be consistent and up to date with the current state of the system.
2. When the sensor in the internal area detects an object with the size of a car for too long time the sensor sends a message {ZZ} to all nodes.
3. When a node had sent the message in the first situation or the internal sensor in the second situation they wait for acknowledgement from all other nodes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Format** | **Description** | **Content** |
| Current node | {X} | It can be one of the following characters:   * ‘N’ – North * ‘W’ – West * ‘S’ – South * ‘E’ – East * ‘F’ – Free (i.e., the flag is not taken by any node) | The node that currently holds the flag for the Internal area of the intersection and can freely be green. |
| Next node | {Y} | It can be one of the following characters:   * ‘N’ – North * ‘W’ – West * ‘S’ – South * ‘E’ – East   ‘F’ – Free (i.e., the flag is not taken by any node) | The node that requests to be the next to hold the flag for the Internal area. E.g., in S04 when a car is detected as input in N, but the Current node is W, N becomes the Next node. |
| Zero-zero message | {Z}{Z} | ZZ is the abbreviation for zero-zero. It is a message that is sent by the sensor in the internal area to all nodes when it detects an object with the size of a car for too long time. This indicates that there is a possible blockage in the area that prevents the normal traffic flow. | The message that is sent by the sensor in the internal area to all nodes when it detects an object with the size of a car for too long time. E.g., in S09 when the sensor in the internal area detects a car that is stuck for some time, it sends a message {ZZ} to all nodes, which turns all nodes red. |
| Acknowledgement | {A}{K} | AK is the abbreviation for acknowledgement. It is a message that is sent by a node to another node to confirm that it has received and understood its message. For example, if the West node sends a message {WN} to the other nodes, the other nodes will reply with {AK} to acknowledge the message. This way, the West node knows that the other nodes are aware of the current state of the system and can proceed with the next step. AK is composed of two letters: A and K. A stand for acknowledgement and K stands for OK. The table below explains the meaning of each letter in the message | The acknowledgement message that is sent as a response to a message consisting of Current node and Next node by another node. E.g., in S01 when a car is detected as input in W, W sends a message with the updated Current node and Next node and waits for acknowledgements from all other nodes to continue. |
| Heartbeat message | {H}{B} | HB is the abbreviation for heartbeat. It is a message that is sent by a node to another node to indicate that it is alive and functioning. For example, if the North node sends a message {HB} to the East node, the East node will reply with {AK} to acknowledge the message. This way, the North node knows that the East node is still active and reachable. HB is composed of two letters, H stands for heart and B stands for beat. The table below explains the meaning of each letter in the message | The heartbeat message that is sent periodically by a node to another node to check its status. E.g., in S02 when the North node sends a message {HB} to the East node every T time, the East node replies with {AK} to confirm that it is still alive. |
| Failure message | {F}{L} | FL is the abbreviation for failure. It is a message that is sent by a node to all other nodes to notify that it has detected a node failure in the system. For example, if the South node does not receive a message {HB} from the West node for more than K \* T time, it assumes that the West node has failed and sends a message {FL} to all other nodes. This way, the other nodes are informed of the node failure and can take appropriate actions. FL is composed of two letters: F and L. F stands for failure and L stands for lost. The table below explains the meaning of each letter in the message | The failure message that is sent by a node to all other nodes when it detects a node failure in the system. E.g., in S03 when the South node does not receive a message {HB} from the West node for more than K \* T time, it sends a message {FL} to all other nodes, which makes all nodes go blinking. |
| Recovery message | {R}{C} | RC is the abbreviation for recovery. It is a message that is sent by a node to all other nodes to notify that it has detected a node recovery in the system. For example, if the West node recovers from a failure and sends a message {HB} to the South node, the South node will reply with {AK} to acknowledge the message and send a message {RC} to all other nodes. This way, the other nodes are informed of the node recovery and can resume the normal operation of the system. RC is composed of two letters: R and C. R stands for recovery and C stands for back. The table below explains the meaning of each letter in the message | The recovery message that is sent by a node to all other nodes when it detects a node recovery in the system. E.g., in S04 when the West node recovers from a failure and sends a message {HB} to the South node, the South node sends a message {RC} to all other nodes, which makes all nodes stop blinking. |

Table 17. Message protocol

## Design rationale

Our traffic light system is designed to be efficient and smart. Efficient in its flow through rate and smart by deciding on the order while the cars are approaching the system. We decided to put safety first as that is the most important choice in making a traffic system. But we also decided to try and minimize overall waiting times and traffic density. We accomplish this by implementing a queue system that lets cars pass through at a rate that is determined by an algorithm. This will eventually lead to fewer emissions. These 2 points are the main reason why we are building our system.