FRI-2G.302-2-CCT2-04

BUILDING A CENTRALISED SMART CITY SYSTEM FOR URBAN MOBILITY MANAGEMENT AND SOLVING PROBLEMS RELATED TO PARKING AREAS, PUBLIC TRANSPORT AND ECO-TRANSPORT - TYPES OF COMMUNICATION AND DATA EXCHANGE PROTOCOLS BETWEEN DEVICES IN THE EXTERIOR PART OF THE SMART CITY SYSTEM[26]

Eng. Ivan Kolev, PhD
Department of Telecommunications,
University of Ruse "Angel Kanchev", Bulgaria
Tel.: +359 899 075 092
E-mail: ikolev@uni-ruse.bg

Prof. Georgi Hristov, PhD
Department of Telecommunications,
University of Ruse "Angel Kanchev", Bulgaria
Tel.: +359 82 888 353
E-mail: ghristov@uni-ruse.bg

Assoc. Prof. Plamen Zahariev, PhD
Department of Telecommunications,
University of Ruse "Angel Kanchev", Bulgaria
Tel.: +359 82 888 353
E-mail: pzahariev@uni-ruse.bg

Abstract: *The focus of this paper is to present the basic protocols for communication and all the building blocks of the entire hardware and software structure in the construction of the Smart City System for public transport. The basic concepts of the names and the function of the individual building elements and their division into hardware and software parts of the whole structure of the two main subdivisions will be introduced, conditionally called Exterior and Interior. The need to create such a centralized system, which consists of many different software applications communicating via API (Application Programming Interface), data collection in the central database, performing the necessary computational actions on central servers will be considered. The structure of the communication protocols for data exchange via REST API, which are specifically developed for Smart City System - Public Urban Transport, will also be considered and the overall concept of the data exchange between the individual hardware and software modules will be explained through these protocols.*

Keywords: *Smart City, smart solutions, public transport, eco transport, LoRaWAN network, API and central database, smartphone app, Android, iOS, Validators, Centralized system, Web applications, Servers, efficiency, GPS*

INTRODUCTION

The main communication process, and respectively the one with the largest number of commands from the communication protocol, is the one between the Driver's Terminal and the Central Server (Lim, C., Kim, K., & Maglio, P., 2018). It should be noted that the system must be designed to work in different configuration variants, respectively with and/or without the presence of certain hardware from the EXTERIOR.

[26] Докладът е представен на сесия на секция 3.2 на 28 октомври 2022 с оригинално заглавие BUILDING A CENTRALISED SMART CITY SYSTEM FOR URBAN MOBILITY MANAGEMENT AND SOLVING PROBLEMS RELATED TO PARKING AREAS, PUBLIC TRANSPORT AND ECO-TRANSPORT - TYPES OF COMMUNICATION AND DATA EXCHANGE PROTOCOLS BETWEEN DEVICES IN THE EXTERIOR PART OF THE SMART CITY SYSTEM

Configuration options are as follows:

- **Configuration only with Validator operating in MASTER mode -** in this variant, the driver does not have a control terminal, and in its place is the validator itself and the validator can take over most of the main functions of the Driver's Terminal, respectively the communication with the central server must be performed by the Validator. In this case there is no 4G router, and the SIM card for Internet communications is in the Validator itself.
- **Configuration without Driver terminal with 1 validator operating in MASTER mode and with 1 or more validators in SLAVE mode located in the passenger compartment** - in this case, there is a 4G router, which is the link between the SLAVE Validators and the MASTER Validator, which communicates with them and with the Central Server.
- **Configuration only with Driver Terminal without the presence of a Validator** - in this variant, the Driver's Terminal itself must be able to perform the functions of a Validator, in addition to its own functions. In this case there is no 4G router, and the SIM card for internet communication is in the Driver's Terminal itself. The driver's terminal must have not only a built-in camera for validating the QR codes of the PD, but also a built-in NFC reader for reading plastic subscription cards.
- **Configuration with MASTER Driver Terminal and 1 or more SLAVE Validators located in the passenger compartment** - in this case, all communication with the Central Server is carried out by the Driver's Terminal, as the connecting unit in the vehicle remains a 4G router between it and all Validators operating in SLAVE mode. The advantage of having a Driver's Terminal is first of all driver convenience, easy operation of touch screen menus, visual feedback, messages from validators for irregularities, possibility for navigation system, possibility for messages from the monitoring centre.
- **Advanced configuration with Driver Terminal in MASTER mode and with 1 or more Validators in SLAVE mode -** this option is exactly the same as the previous one, with the addition that the terminal has a connection of a fiscal device and when selling paper tickets, the sale of the fiscal device is registered and a receipt is printed. There may also be no paper tickets, and the receipt itself must be a valid PD (Transport Document), with the necessary information printed on it.

EXPOSITION

The focus of this article is on the structural elements (hardware and software modules) building the EXTERIOR of a Smart City system in Public Urban Transport. As already mentioned in a previous related paper, the EXTERIOR includes all the hardware elements and the software that controls them, which are located outside of a Centralized System. This is all the equipment in each city transport vehicle, the software and the communication between the Driver Terminal and the Central Server, between the Validator and the Central Server, and we tentatively called these protocols "External Data Exchange Protocols" in the Smart City system of the Public City Transport.

Here we can also make the following conditional division:

- **Communication protocols Driver Terminal and Central Server;**
- **Communication protocols Validator and Central Server;**
- **Control protocols Control terminal and Central Server;**
- **Communication protocols Client application and Central Server.**

We have already mentioned that the main external protocols are those with which the hardware in the respective vehicle communicates with the control panel, namely as follows we have the following commands:

Device registration in the system - this is the command with which every time when starting the device (Driver Terminal or Validator), they declare their ONLINE presence and operability to the Central System. It should be noted here that with a new device, with software just installed, there is still no registration in the system and when it is first started, it will automatically register in the system, whereby a unique UUID will be returned to it from the system. must be stored by the device itself. At this initial registration, the device is not yet ready for operation until it is approved and accordingly associated with the vehicle by an administrator in the central system. At each subsequent start, the device is already identified by its UUID and all subsequent events are already associated with this device / vehicle. The registration is completed using the following URL and commands:

REST API URL: https://xxx.xxx.xxx.xxx/api/register-device

Request:{"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","position":{"latitude":43.826592,"longitude":25.9460364,"altitude":0,"accuracy":15.35099983215332},"device":{"device_type":2,"uuid":"7034050ff58c67d1","manufacturer":"HUAWEI","platform":"Android","model":"AGS-L09","version":"7.0","isVirtual":0,"serial":"HEKNU17731201640","cordova":"9.0.0","id":45,"vehicle_idd":1,"account_id":1},"settings":{},"scanner":{},"user":{},"data":{}}

Response:{"code":"ok","message":"ok","device":{"device_type":2,"uuid":"7034050ff58c67d1","manufacturer":"HUAWEI","platform":"Android","model":"AGS-L09","version":"7.0","isVirtual":0,"serial":"HEKNU17731201640","cordova":"9.0.0","id":45,"vehicle_idd":1,"account_id":1}}

Login to the system - devices in public transport vehicles must have a basic driver identification (Fig. 1). As already mentioned in the configurations with available Driver Terminal + Validator or only with Validator we have driver identification through his personal MIfare identification card, which he scans to the NFC reader of the Validator and in the presence of Driver Terminal a command is sent from the Internal Protocols and is also logged on to the Driver's Terminal. The URL and the commands are given below:

REST API URL: https://xxx.xxx.xxx.xxx/api/login

Request:{"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","position":{"latitude":43.826592,"longitude":25.9460364,"altitude":0,"accuracy":15.35099983215332},"device":{"device_type":2,"uuid":"7034050ff58c67d1","manufacturer":"HUAWEI","platform":"Android","model":"AGS-L09","version":"7.0","isVirtual":0,"serial":"HEKNU17731201640","cordova":"9.0.0","id":45,"vehicle_idd":1,"account_id":1},"scanner":{},"user":{"username":"DRIVER","password":""}}

Response:{"code":"ok","message":"ok","access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","user":{"id":1,"account_id":1,"workplace_id":1,"office_id":null,"first_name":"Иван","middle_name":"Чавдаров","last_name":"Колев","email":"ivan@nrjsoft.com","email_notifications":""},"departments":[{"id":1,"department":"T1H"},{"id":2,"department":"T2"},{"id":3,"department":"T2H"},{"id":10,"department":"T9"},{"id":14,"department":"T13"},{"id":20,"department":"T21"},{"id":22,"department":"T24"},{"id":23,"department":"T26"},{"id":24,"department":"T27"},{"id":26,"department":"T29"}]}

When we only have a Driver Terminal, this can be done in 2 ways, respectively by manually entering a User / Password from the screen (Fig. 1) or at a terminal with a built-in NFC reader again by scanning the MIfare ID card. Authorization in both cases is done ONLINE via a command to log in to the system. In the response from the server, in addition to confirmation of successful authorization, only a list of only allowed Departments (Routes from the public transport network) for the specific driver is returned, and are displayed on the initial screen after Login (Fig. 2).

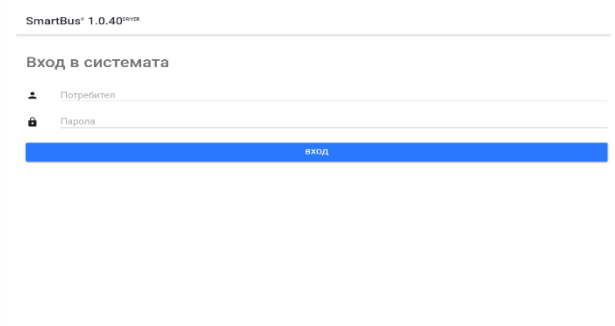

Fig. 1. The login screen of the system

Fig. 2. The screen for the selection of the trolleybus line

Retrieve sub-schedules - each Department (Route) that is known to public transport users / passengers under alpha numeric codes for example: A1, A8, A12 (for bus lines) and T2, T13, T27

(for trolleybus lines) in the system is defined by its 2 routes conditionally called Going - from the Begin bus stop to the End bus stop and Return - from the End bus stop to the Begin bus stop. Apart from the 2 routes, which make up the set of intermediate points / stops between the start and end stop, the corresponding schedule is associated with one route line. The schedule is defined as a set of hours / minutes of arrival of a vehicle at a given stop on the respective route. In other words, the timetable defines all the moments in which a given vehicle moving on this line must be at a given point/bus stop at a specific hour / minute within 1 day. The busier a Route Line, the more often there must be a Bus / Trolleybus running on it, for each point/bus stop, which means that on this line 1,2,3 should move one after the other, Vehicles, turning the full cycle on the Go/Return routes, after which the cycle repeats again. It is for this reason that the entire time table can be conditionally divided into several sub-timetables, just as many as there will be vehicles driving on this line. For example, if 4 separate vehicles have to travel and service on line T13, then the main schedule will be conditionally divided into 4 sub-schedules, so that moving at a certain time interval, they will be at a certain point in the schedule at a given point / bus stop. With this command, after selecting a Route Line, the list of sub-schedules is displayed:

REST API URL: https://xxx.xxx.xxx.xxx/api/get-sub-schedules

Request:{"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","device":{"device_type":2,"id":45,"vehicle_idd":1,"account_id":1},"settings":{"account_id":1,"user_id":1,"department_id":"14","department":"T13","schedule_type_id":0,"schedule_type_name":0,"sub_schedule_id":0,"route_id":0,"schedule_id":0,"start_time":0,"schedules":[],"routes":[],"vehicle_id":1}}

Response: {"code":"ok","message":"ok","schedule_type_id":3,"schedule_type_name":"Зимно съботно", "subschedules": {"1":{"99":[{"route_id":99,"line_name":"T13: Обръщало-кв. Дружба - Гара Разпределителна", "schedule_id":2993, "sub_schedule_id" :1,"start_time": "05:55","disabled":"disabled"},{"route_id":99,"line_name":"T13: Обръщало-кв. Дружба - Гара Разпределителна" ,"schedule_id":2995, "sub_schedule_id":1, "start_time":"06:55" ,"disabled":"disabled"} , {"route_id":99,"line_name" :"T13: Обръщало-кв. Дружба - Гара Разпределителна", "schedule_id":2997, "sub_schedule_id":1,"start_time":"07:55","disabled":"disabled"},{"route_id":99,"line_name":"T13: Обръщало-кв. Дружба - Гара Разпределителна", "schedule_id":2999,"sub_schedule_id":1,"start_time":"09:05","disabled":"disabled"}],"100":[{"route_id":100,"line_name":"T13: Гара Разпределителна - Блок №28","schedule_id":3094,"sub_schedule_id":1, "start_time":"06:25", "disabled":"disabled"},{"route_id":100,"line_name":"T13: Гара Разпределителна - Блок №28", "schedule_id":3096,"sub_schedule_id":1,"start_time":"07:25","disabled":"disabled"},{"route_id":100,"line_name":"T13: Гара Разпределителна - Блок №28","schedule_id":3098,"sub_schedule_id":1,"start_time":"08:25","disabled":"disabled"}]}}}

Choice of current schedule - after the driver starts his shift and has successfully entered the system, he selects a Department (Route Line) once for his work shift, after which he is displayed with a selection of all sub-schedules together with hours/minutes of departure on the respective routes and selects his one-time sub-schedule (Fig. 3). With this choice at the beginning of the change of each Driver, the system already "knows", according to the given Line / Route / Schedule, which vehicle, driven by exactly which driver is moving/serving. This is the command that registers Driver's choice:

REST API URL: https://xxx.xxx.xxx.xxx/api/change-schedule

Request:{"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","device":{"device_type":2,"id":45,"vehicle_idd":1,"account_id":1},"settings":{"account_id":1,"user_id":1,"department_id":"14","department":"T13","vehicle_id":1}}

Response: {"code":"ok","message":"Успешно зададохте Разписание 2 за Маршрутната линия T13","subschedule":[{"schedule_id":2994,"schedule_type_id":3,"schedule_name":"1","start_time":"06:25","end_time":"06:49","route_id":99,"sub_schedule_id":2}],"routes":{"99":{"route_id":99,"route_status":1,"line_name":"T13: Обръщало-кв. Дружба - Гара Разпред елителна","department_id":14,"subschedule" :[{"schedule_id":2994, "schedule_type_id" :3, "schedule_name":"1" ,"start_time" :"06:25","end_time":"06:49","route_id":99, "sub_schedule_id":2, "disabled":"disabled"}]}, "100":{"route_id":100, "route_status ":1,"line_name":"T13: Гара Разпределителна - Блок №28", "department_id":1, subschedule" :[{"schedule_id": 3, "schedule_type_id":3,"schedule_name":"1", "start_time":"06:55","end_time":"07:21","route_id":100,"sub_schedule_id":2, "disabled":"disabled"}]}},"department_id":"14","department":"T13","sub_schedule_id":"2","route_id":99,"schedule_id":3006}}

Start of a course from a schedule on a given route - as we have already noted after selecting the relevant sub-schedule, the driver must follow the cycle of traffic on the respective routes for the selected line, striving to follow this schedule as accurately as possible, ie. be at the exact moment of the bus stop (Fig. 4).

During the work shift of the driver, this cycle is repeated N times, and the sub-schedules themselves are tailored so that they include mandatory sanitary breaks, after each cycle Go / Return. When a sub-schedule is selected on the screen The Driver Terminal displays the complete information about the current Line / Route / Schedule, as well as all the initial departure times for a given course. The concept of course can be defined as follows: it is a one-time movement on one of the two routes

Go / Return starting at a specific time / minute according to the selected schedule. Here, according to the policy of the management of the respective public transport and / or according to the choice of the specific driver, each course can be started or registered in the system in two ways, Automatic or Manual. The software of the driver terminal allows one or the other method to be set by selecting the settings.

- In the Automatic Start of Course on Route Option, the software monitors the system time of the terminal and when a moment coincides with a given hour / minute of the selected schedule, it is automatically Requested to the central server to register the start of a course on Line / Route / Schedule, a short message is displayed on the terminal announcing the automatic start of a course and a short sound signal is issued. In this case, the driver has no commitment to the system to register his courses in the system as this happens completely automatically.

- In the Manual Route Start option, the software also monitors the system time of the terminal and when a time coincides with a given hour / minute of the selected schedule, a dialog box is displayed on the terminal screen asking if the driver wants to start a course or not. But here a manual confirmation from the driver is required to perform a Request to the central server to register the start of a course on a Line / Route / Schedule (Fig. 5). After confirmation by the terminal driver, a message announcing the start of a course is displayed and a short sound signal is issued.

The command with which the beginning of the course is registered is:

REST API URL: https://xxx.xxx.xxx.xxx/api/start-route-schedule

Request: {"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","settings":{"account_id":1,"user_id":1,"department_id":"14","department":"T13","schedule_type_id":3,"schedule_type_name":"Зимно съботно","sub_schedule_id":"2","route_id":"99", "schedule_id":"38", "start_time":"13:55","plate_number":"P8888AP","owner_name":" Колев ", "email":"iv_kolev@mail.bg" ,"phone":"8888888888",,"auto_start_course":false,"front_camera":true,"back_camera":false,"vehicle_id":1}}

Response: {"code":"ok","message":"Стартиране на курс по Маршрут с начален час 13:55","currroute": {"department_id":"14","department":"T13","route_id":"99","schedule_id":"3008","start_time":"13:55","course_id":"1"}}

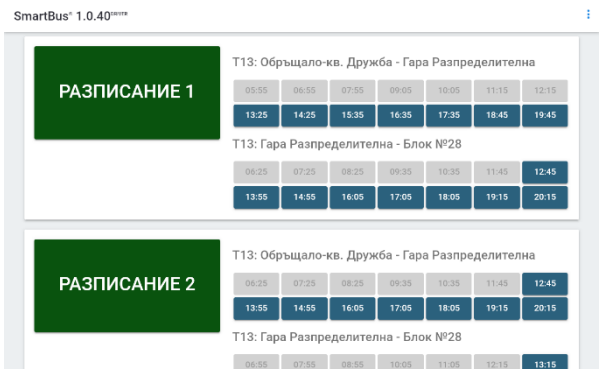

Fig. 3. The bus schedule selection screen

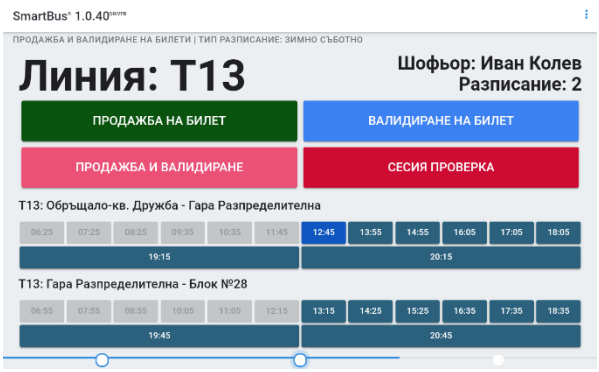

Fig. 4. The screen for selection, purchase and validation of the tickets

Ticket sale through the driver's terminal - through the driver's terminal there is a possibility to sell and / or validate paper tickets. Depending on the selected configuration, as mentioned above with or without a local fiscal device, respectively when selling pre-printed cob tickets with a unique QR code for each, a fiscal device is not necessary as the ticket itself is a security for which the necessary taxes and fees have been prepaid to the state in advance. In the other case, the receipt itself

printed by the local fiscal device may be a PD. In both cases, the sale of a Paper Ticket is registered in the centralized system with the following command:

REST API URL: https://xxx.xxx.xxx.xxx/api/ticket-purchase

Request: {"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","device":{"device_type":2,"id":45,"vehicle_idd":1,"account_id":1},"settings":{"account_id":1,"user_id":1,"department_id":"14","department":"T13","vehicle_id":1}}

Response: {"code":"ok","message":"Успешна продажба на билет"}

Save driver terminal settings - the driver terminal has the ability to set some settings, such as basic information about the vehicle itself - required during the initial registration to identify the device in the system and its association with the vehicle (Fig. 6). Also, for using a front or rear camera to scan QR codes during validation, as well as to automatically or manually start a course on a route. The settings are registered in the system with the following command.

REST API URL: https://xxx.xxx.xxx.xxx/api/settings-bus-driver

Request: {"access_token":"ff2a2f498d7f4af1b05137a11ac70e6d","position":{"latitude":43.826592,"longitude":25.9460364,"altitude":0,"accuracy":15.35099983215332,"speed":0,"timestamp":1635584015477},"device":{"device_type":2,"uuid":"7034050ff58c67d1","manufacturer":"HUAWEI","platform":"Android","model":"AGS-L09","version":"7.0","isVirtual":0,"serial":"HEKNU17731201640","cordova":"9.0.0","id":45,"vehicle_idd":1,"account_id":1},"settings":{"account_id":1,"user_id":1,"department_id":"14","department":"T13","schedule_type_id":3,"schedule_type_name":"Зимно съботно","sub_schedule_id":"2","route_id":"99", "schedule_id":"3008", "plate_number":"P8888AP","owner_name":" Иван Колев ", "email":"iv_kolev@mail.bg", "phone":"8888", "advanced_mode":true,"auto_start_course":false,"front_camera":true, "back_camera":false,"vehicle_id":1}}

Response: {"code":"ok","message":"Данните са запазени успешно","setdriver":{"account_id":1,"user_id":1, "department_id":"14", "department":"T13","schedule_type_id":3,"schedule_type_name":"Зимно съботно","sub_schedule_id":"2" ,"route_id":"99", "schedule_id":"3008", "start_time":"13:55" "schedules":[], "routes":[],"plate_number":"P8888AP","owner_name":" Иван Колев "email":"iv_kolev@mail.bg","phone":"888,"auto_start_course":false,"front_camera":true,"back_camera":false,"vehicle_id":1}}

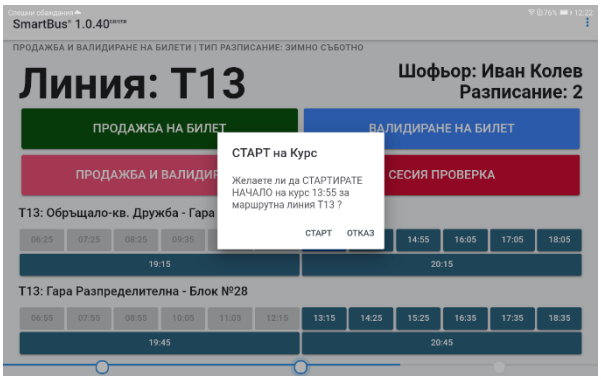

Fig. 5. The interactive message for confirmation of the selected bus line trip

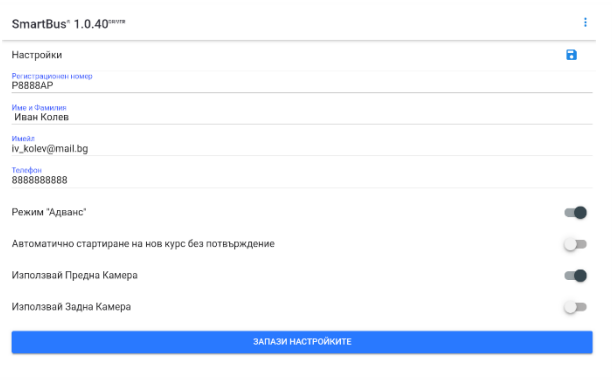

Fig. 6. Screen for configuration and saving of the vehicle and driver parameters

CONCLUSION

The application of smart technologies is increasingly finding its place in the transport and logistics sector, and here in particular we have briefly considered a part of this sector of high public interest, such as Public Urban Transport (Souza, A., Figueredo, M., Cacho, N., Araújo, D., & Prolo, C., 2016).

Smart City systems in public transport can be implemented in many different ways depending on the purpose and requirements (Hashem, I., Chang, V., Anuar, N., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., & Chiroma, H., 2016 and Jara, A., Genoud, D., & Bocchi, Y., 2014), as well as depending on the technical preferences of hardware and software engineers in the design itself to meet the relevant requirements (Souza, A., Figueredo, M., Cacho, N., Araújo, D., & Prolo, C., 2016). Due to its great functionality and complexity, such a system is composed of many separate elements, respectively hardware and software modules, some of them are in a static environment such as the control centre, and others are located in the vehicles themselves and are in constant motion to have

increased requirements for resistance of any kind during their operation. Therefore, many different technologies are used in such a system, both software and hardware, and they must be carefully selected by design engineers at the design stage to ensure high reliability and long life at a later stage for a period of operation at minimal cost for maintenance. Only under these conditions would the system be efficient and reliable and would have a high return on initial investment and therefore a profit (Thakuriah, P., Dirks, L., & Keita, Y., 2016).

The benefits of using an automated system for sales, reporting and validation of travel in urban transport are huge and nowadays it is no longer a question of whether such a system is needed in the public urban transport of any municipality, be it small or large, but when and how it would be realized.

ACKNOWLEDGMENTS

The publication is developed with the support of Project BG05M2OP001-1.001-0004 UNITe, funded by the Operational Programme “Science and Education for Smart Growth”, co-funded by the European Union trough the European Structural and Investment Funds.

This publication reflects the results obtained under Project 2022-FEEA-03 "Development of a robotic autonomous platform for obtaining and analysing spectral images of the ground surface", financed by the Scientific Research Fund of the University of Ruse "Angel Kanchev".

REFERENCES

Lim, C., Kim, K., & Maglio, P. (2018). *Smart cities with big data: Reference models, challenges, and considerations*, Cities, Vol. 82, 2018, Pages 86-99, ISSN 0264-2751, https://doi.org/10.1016/j.cities.2018.04.011.

Hashem, I., Chang, V., Anuar, N., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., & Chiroma, H. (2016). *The role of big data in smart city*, International Journal of Information Management, vol. 36, no. 5, pp. 748–758, 2016.

Souza, A., Figueredo, M., Cacho, N., Araújo, D., & Prolo, C. (2016). *Using big data and real-time analytics to support smart city initiatives*, IFAC-PapersOnLine, Vol 49, Issue 30, 2016, Pages 257-262, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2016.11.121.

Thakuriah, P., Dirks, L., & Keita, Y. (2016). *Digital Infomediaries and Civic Hacking in Emerging Urban Data Initiatives*, Seeing Cities Through Big Data: Research Methods and Applications in Urban Informatics, New York, NY, USA, Springer, Oct. 2016, pp. 189–207.

Jara, A., Genoud, D., & Bocchi, Y. (2014). *Big data in smart cities: From poisson to human dynamics*, 28th International Conference on Advanced Information Networking and Applications Workshops, May 2014, Victoria, Canada, pp. 785–790.