

Web of Things Based Advanced Smart Parking Management Solution

Luciano Alessandro Ipsaro Palesi, Matteo Naldi, and Paolo Nesi(✉)

Department of Information Engineering, Distributed Systems and Internet Technology Lab (DISIT), University of Florence, Via Santa Marta, 3, 50139 Florence, Italy
paolo.nesi@unfi.it

Abstract. Urbanization and increasing vehicle ownership have exacerbated parking challenges in cities worldwide, necessitating innovative solutions for efficient space utilization. Smart Parking Management Systems, integrating the Internet of Things (IoT), Web of Things (WoT), machine learning, and digital twins, offer a data-driven approach to optimizing parking infrastructure. This paper presents an advanced Smart Parking Management System developed using Snap4City, an open-source framework designed for real-time urban mobility monitoring. The proposed solution enables real-time parking occupancy tracking, predictive analytics, and automated enforcement, improving overall efficiency, sustainability, and user experience. Through dynamic pricing models, integration with Mobility-as-a-Service (MaaS), and AI-driven forecasting, the system enhances urban mobility while reducing traffic congestion and environmental impact. The effectiveness of the solution is validated through simulations and implementation in Florence, demonstrating its capability to streamline parking operations, support municipal policies, and improve user accessibility. The platform has been implemented using data from Florence and is built on the Snap4City Open Source platform for CN MOST, national center on sustainable mobility.

Keywords: Smart Parking Management · Parking Policy · IoT · WoT · smart applications

1 Introduction

Urbanization is an ongoing global trend, with 53% of the world's population currently residing in cities that occupy only 2% of the Earth's surface [1]. The rapid increase in the number of vehicles on urban roads has led to rising concerns regarding parking space management. Efficiently utilizing available parking areas is crucial to accommodate the growing demand. The difficulty in finding parking spaces negatively impacts both transportation efficiency and sustainability [2]. Searching for a parking spot contributes to unnecessary traffic congestion, increased fuel consumption, and higher emissions, leading to environmental degradation [3]. These issues are particularly relevant when considering on-street parking, but they also extend to off-street parking facilities, which may experience unpredictable occupancy fluctuations depending on time, location, and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15896, pp. 432–450, 2026.
https://doi.org/10.1007/978-3-031-97654-4_27

external factors unknown to drivers. According to a study [4] drivers spend approximately 15 minutes searching for a suitable parking space, further exacerbating urban mobility issues. The increasing demand for parking spaces within urban areas stems from various factors, including urban expansion, rising vehicle ownership, insufficient urban planning, and inefficient parking management systems.

Smart parking management systems based on digital twins, represent a significant advancement in optimizing urban transport infrastructures. They integrate technologies of Internet of Things (IoT), Web of Things (WoT), and machine learning. The main aim of smart parking is to address parking scarcity, reduce traffic congestion due to reduction of cars looking for parking, and provide a more efficient resource allocation.

Digital twins [5] enable dynamic modelling of parking environments, improving efficiency through:

- **Real-time monitoring and data-driven decision-making:** Digital twins use IoT sensors, such as ultrasonic, magnetic, and camera-based technologies, to collect real-time parking occupancy [6]. The parking status data is processed to ensure accurate availability updates and communicate the status to drivers via mobile apps or digital signages.
- **Predictive analytics and parking demand forecasting:** Machine learning, AI algorithms analyse historical and real-time data to predict parking occupancy and even demand [3]. Predictive analytics may also enable to compute dynamic pricing models according to the demand/offer, and also to compute the optimal fair ensuring efficient parking allocation and revenues [7].
- **Scenario simulation and urban planning:** Digital twin has to provide simulation capabilities to model urban scenarios, allowing planners to assess the impact of different strategies according to What-if analysis, before implementation. The simulations may be used to optimize the positioning of parking meters to ensure maximum coverage, minimum discomfort for using them, and thus increase efficiency. The analysis of traffic flow and parking demand, the solution should provide support on designing more efficient transportation infrastructures and thus to reducing congestion and emissions [8].

For effective Smart Parking Management Systems is essential to implement the above-described features. These systems collect real-time data from IoT sensors, parking payment systems, mobile apps to facilitate the communication between parking infrastructure and management platform. Moreover, technologies cloud/edge computing, and machine learning to make predictions may actually implement smart parking systems with valuable features for both drivers and urban planners.

In this paper, we propose a smart parking management system developed using Snap4City. Snap4City is an open-source framework for creating an enhanced Global Digital Twin [9, 10] integrated with Local Digital Twins, operating across various domains such as mobility [11], security [12], and industry [13]. The objective of this paper is to propose an innovative solution to enhance parking space management, promoting a more efficient use of urban resources while simplifying daily interactions between citizens and local administrations. The proposed system aims to seamlessly integrate into modern urban environments, benefiting both registered users and non-users by improving overall mobility and accessibility. The primary goals of this study include enhancing urban

mobility by improving the driving experience for both residents and visitors, optimizing resource management to reduce traffic congestion and environmental pollution, monitoring parking availability in real-time through data collection and analysis, and modernizing administrative processes to enhance enforcement and regulatory mechanisms for traffic law compliance.

This paper is organized as follows. Section 2 presents the state of the art. Section 3 defines and analyses the requirements for a Smart Parking Management Solution, comparing them with existing approaches. Section 4 introduces the proposed advanced Smart Parking solution. Section 5 describes the Snap4City Smart Parking use case, detailing its implementation. Section 6 details the main interaction diagrams, illustrating the sequence of actions for registered users, non-registered users, and specific system scenarios. Sections 7 and 8 provide validation and an illustrative example based on Florence data, while Sect. 9 presents the conclusions.

2 Related Work

In urban parking management, many commercial solutions [17, 18] focus on stall reservation and monitoring, typically within closed and proprietary systems that often fail to integrate effectively with the complex dynamics of smart cities. Effective parking management extends beyond basic space monitoring and must incorporate advanced functionalities such as automated enforcement of parking regulations and violation penalty management. Additionally, a system should include comprehensive user management, enabling the administration of different user categories—such as residents, disabled drivers, commercial vehicles, and electric vehicle users—to ensure compliance with municipal policies and regulate access to parking spaces.

The management of urban parking represents a critical challenge in the development of sustainable and efficient cities. Existing state-of-the-art proposals focus on improving parking enforcement, user management, and accessibility by leveraging real-time data collection, automated violation detection, and digital permit systems. These efforts aim to enhance urban mobility, simplify interactions between citizens and local administrations, and ensure compliance with municipal parking regulations, ultimately contributing to a more efficient and user-friendly parking experience. In scientific literature, a variety of solutions have been proposed to address the above-described challenges. These solutions of parking management have to include various parking regulations for managing user access and billing.

Murali et al., [14], proposed an IoT-based parking management system to cope with urban parking shortages. The system used sensors to collect real-time data and monitor parking availability, providing user notifications and optimizing space utilization vis mobile app, with limited business models. Venkata et al., [15], introduced an IoT-enabled smart parking system to get real-time data of vehicle entry time, parking duration, slot availability, and payment details. A mobile app updates the users about the availability of free parking slots. Janowski et al., [16], developed a Smart Parking Assistant and Resource Knowledge (SPARK). SPARK utilizes a YOLOv9-based detection system for parking status detection. Moreover, the system primarily focuses on vehicle detection and identifying available parking spaces rather than gathering detailed information

on individual vehicles, which prevents customer identification. Unlike previous Smart Parking Management Systems, Mondal et al., [21] introduced a dynamic pricing model that adjusts parking costs on the basis of the demand. It increases revenue for parking owners and government agencies during peak hours. The solution presents limitations in reliance on real-time data availability for parking occupancy. Additionally, the dynamic pricing model has to be based on efficient IoT info propagation for revenue generation. The dynamic pricing could discourage users when price fluctuations during peak hours become excessive.

3 Ideal Smart Parking Requirements

After analysing the state of the art and considering the needs highlighted in the introduction, an effective parking management system must meet specific functional requirements. Defining these functional requirements is essential for developing scalable and reliable solutions capable of enhancing operational efficiency and optimizing urban parking resources. The following sections outline the key functional requirements for an advanced Smart Parking Management System:

R.1 Coverage of Entities Involved for modelling various parking-related components such as parking lots, on-street spaces, sensors, payment kiosks, digital signage, and enforcement devices. The system should enable the creation, tracking, and management of these entities throughout their lifecycle, including attributes like location, availability status, pricing, and capacity.
R.2 Users' Roles Management to define and control access to the system based on different user profiles. This feature should support multiple roles, such as administrators, enforcement officers, parking operators, residents, and general drivers, each with specific permissions and responsibilities. Proper user access management ensures data security, streamlined workflow, and operational accountability.
R.3 Get real time data from parking spaces, including occupancy status, sensor-based detections, and vehicle entry/exit timestamps. Additionally, the system should track environmental factors such as temperature and air quality within parking structures, as well as potential anomalies (e.g., unauthorized access, obstruction detection, or malfunctioning sensors). These real-time insights enhance parking operations and predictive planning.
R.4 Provide statistical data in terms of Key Performance Indicators (KPI) through interactive dashboards. The KPI dashboards should display real-time insights into parking occupancy at the operators, revenue trends, enforcement efficiency, peak utilization hours, and sensor reliability. By visualizing these KPIs, operators can optimize resource allocation, identify trends, and enhance urban mobility strategies.
R.5 Implement a dynamic parking tariff system that adjusts pricing based on time slots, demand fluctuations, and special conditions (e.g., peak hours, off-peak discounts, and event-based pricing). The tariff structure should be configurable through a Typical Time Trend (TTT) model (defining fair for each hour per day, per week, in a period), allowing municipalities to set minute-based rate variations for each parking facility.
R.6 Manage special permits for different user categories, including residents, disabled drivers, commercial vehicles, and electric vehicle users. The system should facilitate permit issuance, renewal, and validation, ensuring compliance with municipal parking policies and preferential access for eligible users.

R.6.1 Device-Based Permit Management. The system should support the management of special permits through dedicated devices, enabling seamless parking access without requiring manual app interaction. Authorized users, such as residents, disabled drivers, commercial vehicle operators, and electric vehicle users, should be able to initiate parking sessions automatically upon arrival through device recognition.

R.7 Provide alarm management and notification systems to alert operators and users in real time about issues such as overparking, unauthorized parking, device failures, or security breaches. The system should support automated notifications via mobile applications, email alerts, SMS, and web-based platforms, enabling quick responses to potential operational disruptions. Additionally, it should provide content management and user messaging capabilities to ensure timely and relevant notifications. Notifications should include alerts for parking expiration, overparking, fines, unauthorized parking, and wallet balance exhaustion.

R.8 Enforce parking regulations and fine management through automated detection of violations and electronic fine issuance. The system should enable officers to verify violations, issue fines via mobile applications, and integrate with payment systems for fine collection.

R.9 Integration with Mobility-as-a-Service (MaaS), allowing users to combine parking with other transportation options, such as public transit, and bike-sharing. The system should facilitate unified payments, trip planning, and dynamic recommendations based on user preferences, traffic conditions, and public transport availability.

R.10 Smart Parking Routing, provide dynamic parking routing by integrating real-time traffic sensor data, historical occupancy trends, and traffic conditions. The system should guide drivers to the nearest available parking spot, minimizing search time and congestion. It should also adopt recommendations based on user preferences, parking restrictions, and estimated arrival times.

Table 1. Comparison of Parking Management Solutions Based on Requirement Satisfaction

Solution		Requirements										
	R.1	*R.2*	*R.3*	*R.4*	*R.5*	*R.6*	*R.6.1*	*R.7*	*R.8*	*R.9*	*R.10*	*R.11*
Easypark [17]	N	Y	Y	Y	Y	Y	N	Y	Y	N	N	N
Smartparkingsystems [18]	N	Y	Y	Y	Y	Y	N	Y	Y	N	N	N
Murali [14]	N	N	Y	N	N	N	N	Y	N	N	N	N
Venkata [15]	Y	N	Y	Y	N	N	N	N	N	N	N	N
Janowski [16]	N	N	Y	Y	N	N	N	N	N	N	N	N
Mondal [21]	Y	Y	Y	N	Y	N	N	N	N	N	N	N
Snap4CitySmartParking	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

R.11 Predictive Parking Availability to estimate future availability of parking spaces by analysing historical data, real-time occupancy and traffic patterns. The system should allow users to reserve parking spaces in advance, reducing search times and traffic congestion. Predictive models should consider variables such as time of day, weather conditions. By integrating these factors, the system optimizes space allocation and improves overall parking efficiency (Table 1).

4 Advanced Smart Parking Architecture

In Fig. 1, the proposed Advanced Smart Parking Management Architecture is illustrated, highlighting its development on top of the Snap4City [10, 26] open-source framework. The system is designed to manage data ingestion, processing logic, and business intelligence tools for the front end. Snap4City [25] is a Smart City platform offering scalability, multi-tenancy, and the ability to handle millions of daily data entries via protocols such as MQTT and NGSI [1]. It is GDPR-compliant [19] and ensures data protection through a user role management system, providing secure and personalized access. The platform supports Web of Things, enabling flexible entity modelling [20] and managing their entire lifecycle. Data ingestion is handled via Node-RED [10], while an access control mechanism assigns specific roles—administrators, operators, and officer staff—to ensure both security and operational efficiency. In the proposed Advanced Smart Parking, data is collected through IoT sensors [24] (or using technologies such as LoRa, Bluetooth and Narrowband -IoT) deployed across the city and transmitted to NGSI Brokers for processing. The system manages data ingestion, storage, and processing logic via microservices, ensuring efficient data flow and decision-making capabilities. Parking predictions are performed according to [27].

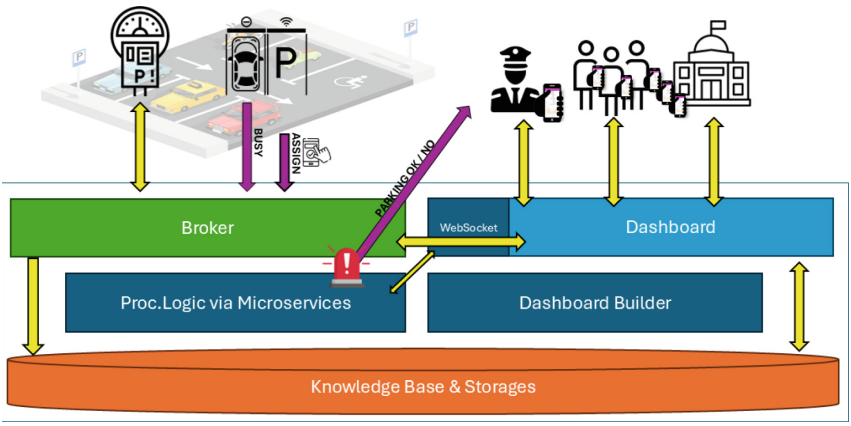

Fig. 1. Advanced Smart Parking architecture. The system integrates sensors, brokers, and dashboards to manage parking. Purple arrows indicate the basic interaction: the sensor detects a parked vehicle, the user confirms the slot number, and the system can notify municipal police if necessary.

Advanced Smart Parking collects and monitors real-time data from:

- **Parking sensors (On-road, Off-road):** Real-time occupancy status, availability, etc.
- **Dashboard/Mobile app for users**: allows users to start and stop parking sessions.
- **Dashboard/Mobile app for operators:** Allows parking enforcement personnel to monitor parking status, check payments, and manage violations.
- **Parking meters**: Enables non-registered users to start a parking session, process payments, and track parking duration.

The application supports a large range of payment methods and business models, which can be customized by area and municipality, allowing both registered and non-registered users to seamlessly start and stop parking sessions. The advanced monitoring and reporting capabilities offered by Snap4City facilitate data-driven policy adjustments, optimizing overall parking management efficiency.

5 Main Smart Parking Use Cases

This section presents the Snap4City Smart Parking use cases, which are illustrated in the next section via sequence diagrams. These use cases highlight the main views and relations with different users and the system, focusing on the roles of the municipal operator, end user, and officer.

5.1 Use Case – Municipal Operator

The municipal operator is responsible for managing and monitoring parking areas in real time using an advanced system of interactive dashboards. In Fig. 2, a use case diagram for the municipal operator is presented. The system architecture includes three main functionalities: monitoring the status of devices on the map, statistical analysis to optimize parking area management, and the automated definition of parking rules with

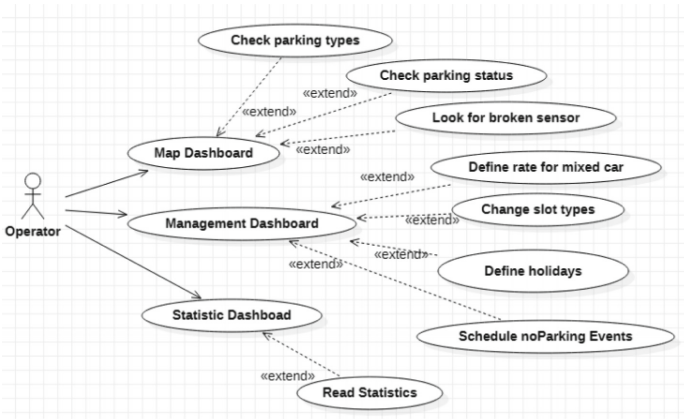

Fig. 2. Use case diagram for Municipal operator.

immediate notification to users in case of changes. The **Map Dashboard** allows operators to view device statuses, check the types of available parking spaces, and identify potential malfunctions, such as faulty sensors. The **Management Dashboard** enables operators to directly intervene in parking rules, modify stall types, define specific rates for mixed vehicles, and schedule extraordinary parking restrictions. Finally, the **Statistic Dashboard** provides a detailed data analysis, offering weekly statistics and distributions by user and vehicle type, facilitating strategic decisions based on quantitative data.

5.2 Use Case – End User, City User

The city user should access and monitoring parking areas status in real time. In Fig. 3, a use case diagram for the end user is presented. The end user, utilizing the parking service, monitors in real-time the parking availability and the economic policies of his/her destination area through a mobile app. The system provides a map overview of the situation, which can be extended with advanced features such as searching for free slots, checking the economic policies of the area, and being routed to the desired slot. Additionally, the user can start and create a "parking event", which implies, the actual parking, to make payments, and receive notifications regarding parking session, enhancing the overall user experience.

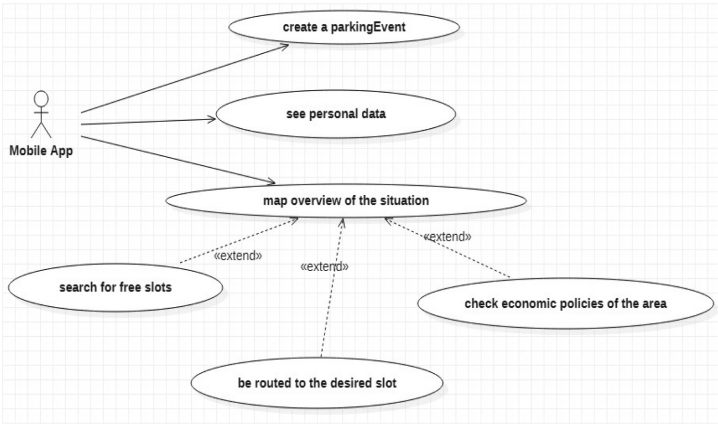

Fig. 3. Use case diagram for End user

5.3 Use Case – Officer

The officer is responsible for ensuring compliance with parking regulations and issuing fines when violations occur. In Fig. 4, a use case diagram for the officer is presented. Through the mobile app, the officer is continuously updated on parking spots where infractions have been detected. The system allows the officer to identity the area with major problems, and check information about the currently parked vehicle, verify the

situation, and, if necessary, issue a fine after conducting the required validations and corrections, finally completing a form with the fine data. This approach enhances the efficiency and accuracy of parking enforcement while ensuring that violations are addressed systematically and in sustainable manner.

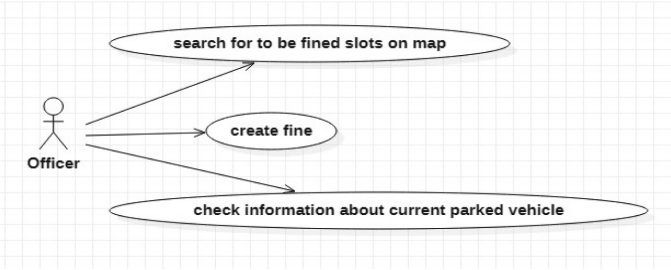

Fig. 4. Use case diagram for the Officer

6 Main Interaction Diagrams of the Solution

This section focuses on the main interaction diagrams of the solution, particularly the sequence diagrams that detail the interactions between different system components and users. These diagrams illustrate the step-by-step processes involved in key scenarios, such as the parking procedure triggered by events, the interactions of registered users with the system, the flow for unregistered users, and the usage of electric charging stations. All the platform is event driven and based on MicroServices. The core microservices to access at data and search for results and creating dashboards and views are those of Snap4City. On the other hand, the complex logic of the Smart parking has been totally developed for this solution.

6.1 Parking Procedure by Events

The process starts with the operation of a parking slot (see Fig. 5). When a vehicle parks, the sensor detects its presence and updates its status to BUSY, sending this information to the IoT application. The server processes this update and adjusts the parking occupancy counts accordingly. Similarly, when the vehicle leaves the parking slot, the sensor updates its status to FREE, and the system updates the counts. When a vehicle arrives at the parking lot and parks, the sensor detects its presence and sends a BUSY notification to the system. If the user (registered or non-registered) starts a parking session via the mobile app or parking meter, the system confirms the session as active. However, if no session is started within a predefined time frame, the system automatically triggers a violation notification to the operator, specifying the parking slot ID where the infraction occurred. In addition to basic parking status updates, the IoT application manages other events, such as charging slot updates, infractions, parking event closures, and mixed-car slot payments. For example, if a recharge slot is in use, the system may update its

status to NO_RECHARGE; in case of an infraction, the system marks it as TOO_LATE; completed parking sessions are set to DONE; and in mixed-car slot scenarios, a wallet event is created for payment processing.

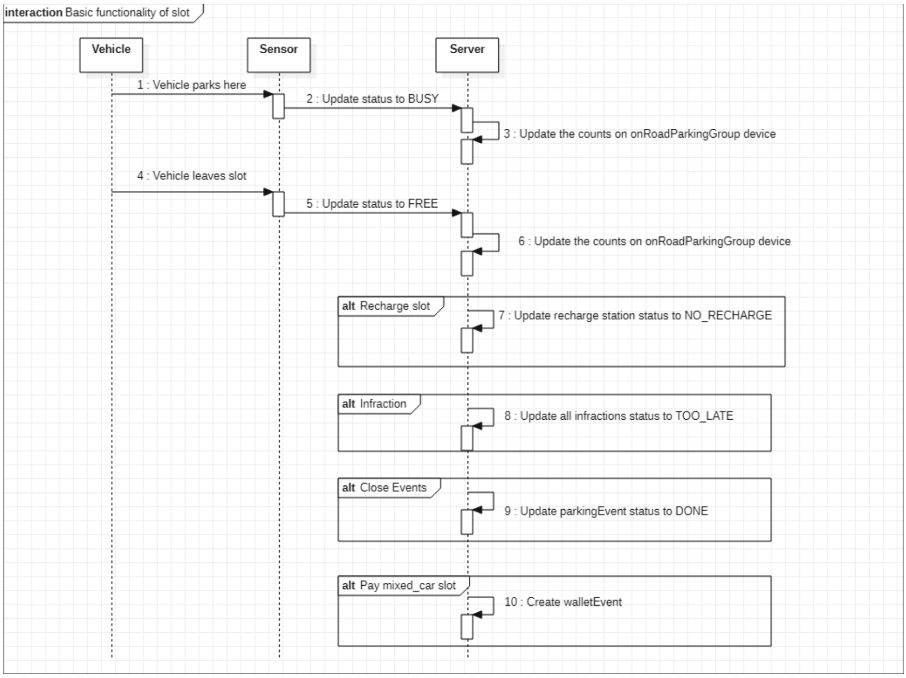

Fig. 5. Sequence of actions illustrating the user's entry and/or exit from the parking slot.

6.2 Parking of a Registered User

The process starts when a registered user with a Bluetooth device or mobile app parks in a designated slot. Upon arrival, the sensor detects the vehicle and updates its status to BUSY, initiating communication with the user's device. This interaction generates a parking event on server side, which is then processed by the system. For users with the mobile app, additional functionalities are available. If the parking slot requires payment, the user can specify a duration (see Fig. 6). In this case, the system sends a renewal notification before expiry. If no duration is set, the parking session automatically ends upon the user's departure. For structured parking facilities, users can request to link their data to a ticket and pay through the app when exiting. Besides standard occupancy updates, the IoT application also manages special events such as infractions, charging slot usage, and mixed-car slot payments. For example:

- If a charging slot is occupied, the system marks it as NO_RECHARGE.
- In case of an infraction, it is flagged as TOO_LATE.

- Completed parking sessions are updated to DONE.

In mixed-car slot scenarios, the system creates a wallet event for payment processing. Furthermore, when a control officer arrives, they input the parking code to retrieve infraction details. For registered users having a special permit equipped with a sensor, once the vehicle is parked, the system automatically updates the parking status. After this update, the parking sensor transmits the user and parking slot identifiers to the system, which records the event as a Bluetooth-based parking session. As shown in Fig. 7, this process allows the system to associate the parking slot with the user's permit without the need for manual input. This mechanism ensures that the parking session is activated

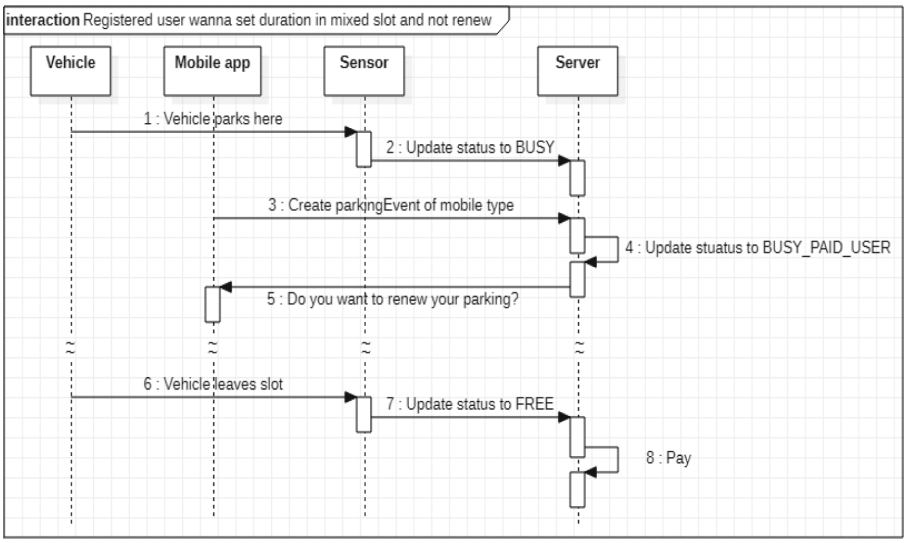

Fig. 6. Sequence of actions illustrating registered user

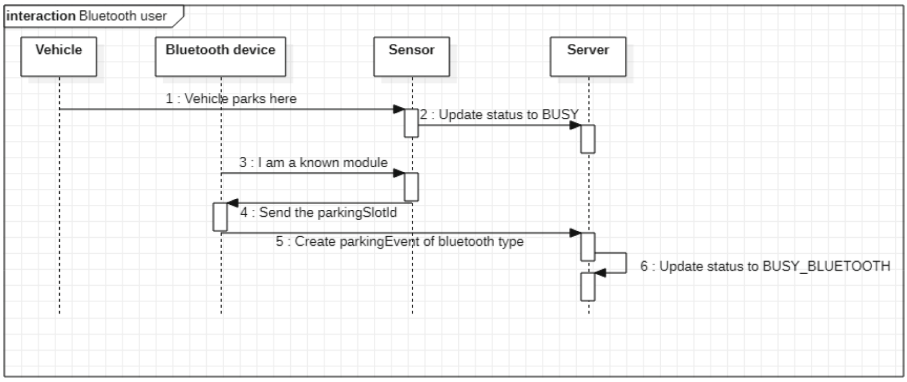

Fig. 7. Sequence of actions illustrating registered user equipped with a sensor

based on sensor detection and that the system can differentiate standard vehicles from those with special permits.

6.3 Parking of a Unregistered User

The process begins when an unregistered user parks in a mixed car slot (see Fig. 8). Upon arrival, the sensor detects the vehicle and updates the slot status to BUSY. The system then waits for a predefined period to check if the user makes a payment. If the user pays at the parking meter, the system updates the slot status to BUSY_PAID_ANON. If no payment is made within the allocated time, the status changes to TO_BE_FINED, and a fine is issued. For cases of overparking, the system detects when the paid duration has ended. The slot status is then updated to BUSY_PAID_ANON_OVER, followed by TO_BE_FINED, triggering a fine for exceeding the paid time. In mixed-car slot scenarios, the system manages anonymous payments and potential infractions. Control officers can check parking details by inputting the slot code to verify fines and infractions.

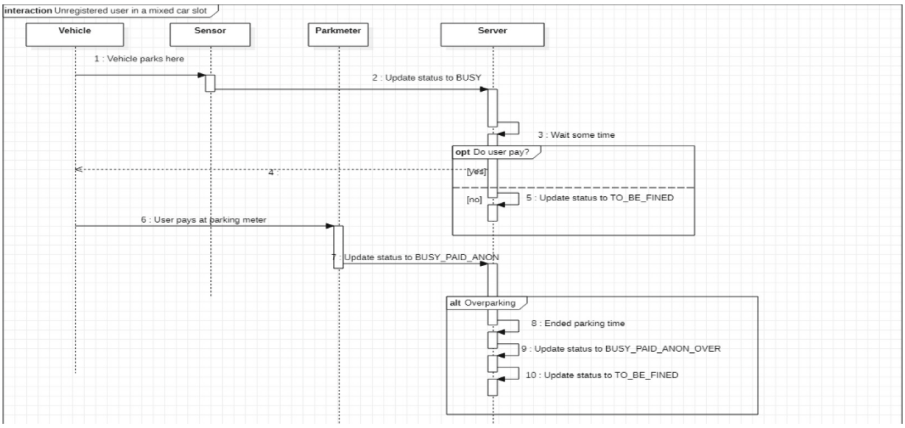

Fig. 8. Sequence of actions illustrating (a) unregistered user entry and/or exit from the parking slot.

6.4 Parking on an Electric Charging Stations

One of the innovations introduced by the project concerns the monitoring of electric charging stations. The goal is to ensure their proper use, preventing vehicles from occupying charging spots without authorization and ensuring that they leave the space once charging is complete. As shown in Fig. 9, the system detects the presence of a vehicle through the sensors and updates the station status to BUSY. If the user does not start charging, the server updates the status to TO_BE_FINED, indicating a possible violation. When recharging begins, the station status changes to RECHARGE_STARTED and, upon completion, updates to RECHARGE_ENDED. If the vehicle remains in the station when charging is complete, the system signals another violation by updating

the status to TO_BE_FINED. Once the vehicle leaves the location, the station status returns to FREE, and if no recharging has occurred, the system sets the status to NO_RECHARGE.

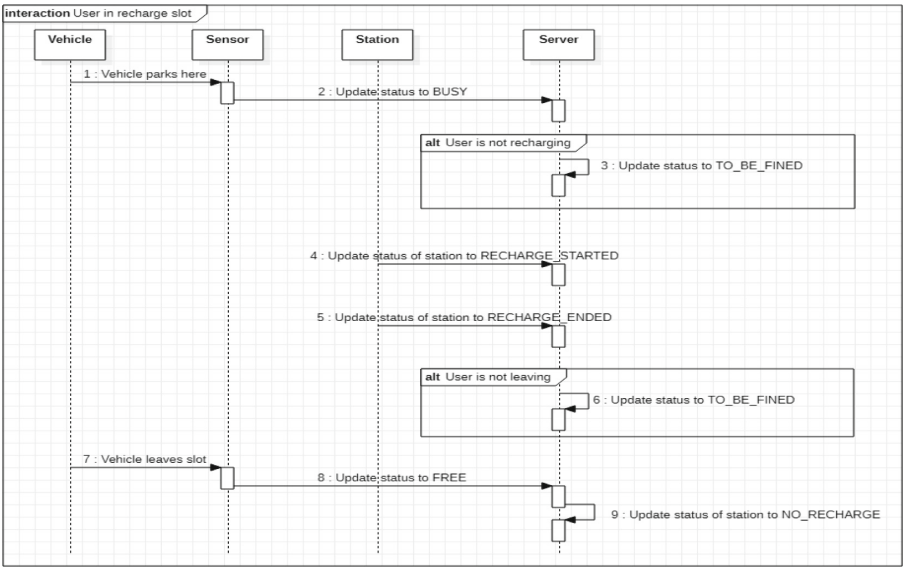

Fig. 9. Sequence of actions illustrating the Operation of an EV Charging Station.

7 Validation via Simulation of Global Parking

To effectively assess the capabilities of the system, a simulation has been implemented, replicating **real-world parking** scenarios. The main execution thread of the simulator (see Fig. 10) is responsible for generating random events (randomEvent()), assigning users (randomUser(event)), and selecting available parking slots (randomSlot(event)). These elements are encapsulated into execution tuples and dispatched to background workers (see Fig. 11) for processing.

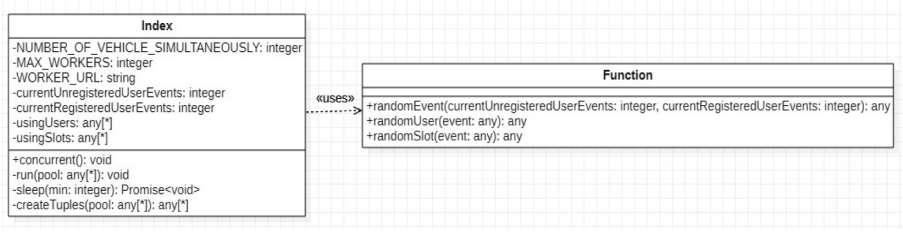

Fig. 10. Class diagram of the main execution thread

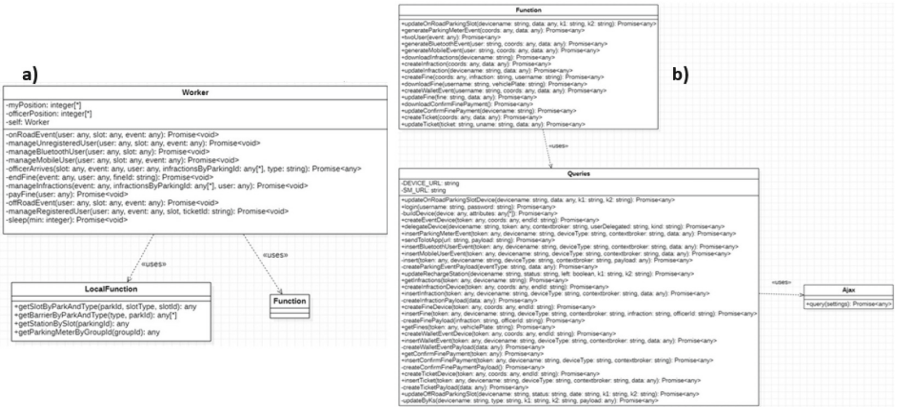

Fig. 11. Class diagram of the Worker class, (a) illustrates the core functionalities of the Worker, (b) expands on additional operations.

The Index class (see Fig. 10) defines key operational parameters, such as NUMBER_OF_VEHICLE_SIMULTANEOUSLY and MAX_WORKERS, which regulate the number of simultaneous parking processes. It employs the run(pool: any[]) function to generate execution tasks. Each task consists of:

- A Worker instance to process the parking event,
- A randomly selected event (randomEvent() from the Function class),
- A user, either retrieved from the database or generated dynamically (randomUser(event)),
- An available parking slot (randomSlot(event)).

If the match from the user (providing slot id and plates) and the busy slot just now (communicating the slot id change of status) is not found, the system iterates until a valid match is assigned to a worker. The tuples generated by the main thread are then dispatched to workers for execution, otherwise means that some critical to be fined conditions has been detected. The workers (see Fig. 11) handle the execution of assigned parking events. The Worker class processes different types of events based on the parking environment. It distinguishes between:

- On-road parking events, managed through the onRoadEvent() function,
- Off-road (structured) parking events, handled by offRoadEvent().
- Each event may involve multiple operations, including:
- Managing users (manageUnregisteredUser(), manageRegisteredUser()),
- Processing mobile and Bluetooth-based interactions (manageMobileUser(), manageBluetoothUser()),
- Handling law enforcement actions, such as detecting violations (manageInfractions()) and processing fines (createFine(), payFine()).

Workers execute these tasks asynchronously, ensuring efficient parallel processing of multiple parking operations. To interact with IoT-enabled devices, the system relies on the Queries and Ajax classes Fig. 11 (b). These components facilitate communication

with external services, such as Snap4City [22], which provides APIs for managing parking sensors, meters, and enforcement devices. Notable API calls include:

- updateOnRoadParkingSlotDevice(), for real-time updates on parking availability,
- generateBluetoothEvent() and generateMobileEvent(), for user interactions via connected devices,
- createFine() and updateFine(), for automated violation management,
- createTicket(), for generating digital parking tickets.

8 Operator Web of Things Interface and Some Examples

For an illustrative example, specific datasets from the city of Florence [23] have been loaded. Three areas of the city of Florence were chosen:

- Piazza Leon Battista Alberti
- The western part of the Statuto district
- The Careggi area

A total of 1,562 on-street parking spaces were identified in these areas, divided into 18 categories:

- **Bus**: Free parking space reserved for buses. A municipal permit is required to park from 08:00 to 20:00.
- **Cargo**: Free parking space for loading/unloading goods. Parking is limited to 15 minutes (08:00–20:00) and requires a permit.
- **Event Car**: Temporary parking space for vehicles during special events (e.g., press for sporting events). A municipal permit is required.
- **Event Moto**: Same as the previous one but for two-wheeled vehicles.
- **Forbidden**: Not an actual parking space but represents areas where parking is prohibited (e.g., in front of driveways). For simplicity, it is considered as a parking space in the project.
- **Free Car**: Free parking space for four-wheeled vehicles.
- **Free Moto**: Free parking space for two-wheeled vehicles.
- **Handicap**: Parking space reserved for disabled persons, requiring a municipal permit.
- **Mixed Car**: Mixed paid parking space for vehicles.
- **Org Car**: Parking space reserved for public sector employees with a municipal permit.
- **Org motorbike**: Same as the previous one but for motorbike.
- **Recharge Car**: Parking space for charging electric vehicles (08:00–20:00). After charging, parking is allowed for a maximum of 60 minutes.
- **Recharge Moto**: Same as the previous one but for motorbike.
- **Resident**: Parking space reserved for local residents with a municipal permit.
- **Taxi**: Parking space reserved for taxis.
- **Timed Car**: Time-limited parking space for vehicles.
- **Timed Moto**: Time-limited parking space for motorbike.
- **Women Car**: Parking space reserved for women, generally located in areas with greater security.

The Snap4City Advanced Smart Parking solution provides a set of views/dashboards dedicated to parking management. One of the main views is Parking Conditions Monitoring view/dashboard in Fig. 12. In this view/dashboard, the operator monitors the real-time status of parking areas and slots throughout the city through an interactive map interface. The system provides an overview of available and occupied parking spaces. Real-time data updates support the operator's decision-making processes. The operator can highlight individual parking spaces according to their status (occupied, free or reserved) or classify areas according to occupancy levels, violations or specific infrastructure elements. The view called Snap4Parking Weekly Statistics (see Fig. 13) reports and allows to perform the analysis on parking occupancy statistics over time, enabling

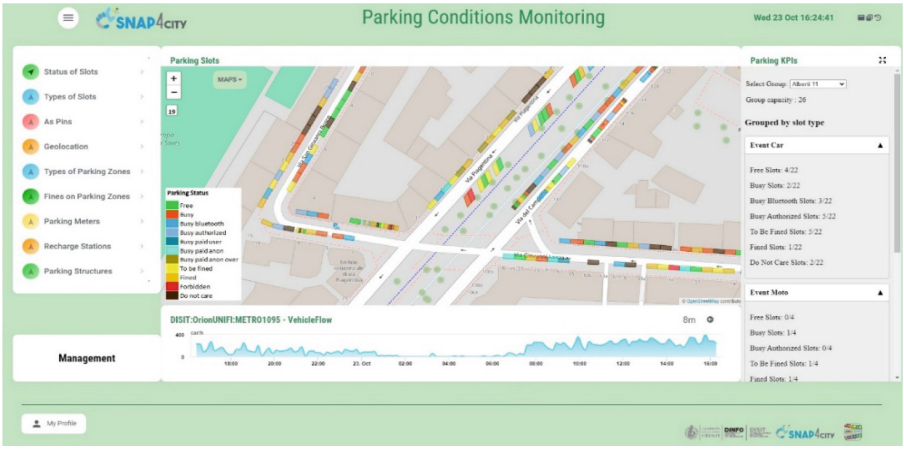

Fig. 12. Dashboard Snap4Parking Conditions Monitoring

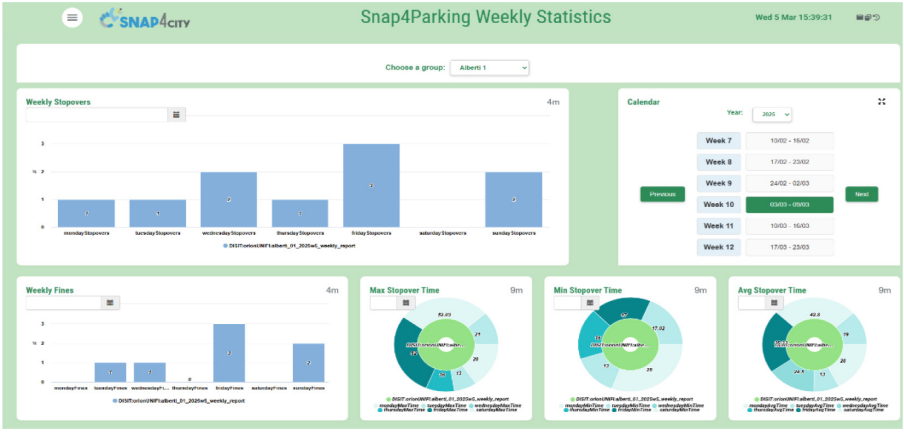

Fig. 13. Dashboard Snap4Parking Statistics

comparisons between different periods. These statistics include key performance indicators such as stopovers, representing short-term parking events, and fines, indicating violations detected within the monitored areas.

9 Conclusions

The effective management of urban parking spaces requires innovative solutions that leverage IoT, data analytics, and real-time monitoring. In this work, a Web of Things-based Advanced Smart Parking Management Solution has been presented, aiming to enhance the efficiency of parking space utilization through real-time data collection and predictive availability. The system enables comprehensive management of parking lots, on-street, and off-street spaces, incorporating role-based access control, dynamic pricing strategies, and integration with Mobility-as-a-Service (MaaS). The simulation conducted using data from Florence demonstrates that the proposed solution can improve the experience of both municipal officers and users. Real-time monitoring facilitates interventions by authorities, while app-based tracking provides drivers with up-to-date availability information, reducing unnecessary traffic and emission. Future research will focus on refining predictive models for parking demand, improving AI-driven optimization algorithms, and exploring integration with emerging urban mobility frameworks. These advancements will contribute to a more sustainable and intelligent urban mobility ecosystem, aligning with the vision of smart city development.

Acknowledgements. This study has been carried out in the context of the Spoke 9 of the Italian National Center for Sustainable Mobility (CN MOST), and the CN MOST flagship project OPTIFaaS and scalability project SASUAM.

References

1. Antoniucci, V., Marella, G.: Small town resilience: housing market crisis and urban density in Italy. Land Use Policy. **59**, 580–588, ISSN 0264-8377 (2016). https://doi.org/10.1016/j.landusepol.2016.10.004
2. Chowdhury, L.H., Mahmud, Z.Z., Islam, I.U., Jahan, I., Islam, S.: Smart car parking management system. In: 2019 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON), pp. 122–126. IEEE (2019)
3. Bilotta, S., Ipsaro Palesi, L.A., Nesi, P.: Predicting free parking slots via deep learning in short-mid terms explaining temporal impact of features. IEEE Access. **11**, 101678–101693 (2023). https://doi.org/10.1109/ACCESS.2023.3314660
4. Mahmud, S.A., Khan, G.M., Rahman, M., Zafar, H.: A survey of intelligent car parking system. J. Appl. Res. Technol. **11**(5), 714–726 (2013)
5. Adreani, L., Bellini, P., Colombo, C., et al.: Implementing integrated digital twin modelling and representation into the Snap4City platform for smart city solutions. Multimed. Tools Appl. **83**, 37121–37146 (2024). https://doi.org/10.1007/s11042-023-16838-0
6. Murali, K., Ramakrishnan, K., Sandeep, S.: Efficient parking management for modern urban environments. In: 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, vol. 2024, pp. 1456–1460. https://doi.org/10.1109/ICACRS62842.2024.10841537

7. Saharan, S., Bawa, S., Kumar, N.: Dynamic pricing techniques for intelligent transportation system in smart cities: a systematic review. Comput. Commun. **150**, 603–625, ISSN 0140-3664 (2020). https://doi.org/10.1016/j.comcom.2019.12.003
8. Canzaniello, M., Amitrano, S., Prezioso, E., Giampaolo, F., Cuomo, S., Piccialli, F.: Leveraging digital twins and generative AI for effective urban mobility management. In: 2024 IEEE Cyber Science and Technology Congress (CyberSciTech), Boracay Island, Philippines, vol. 2024, pp. 146–153. https://doi.org/10.1109/CyberSciTech64112.2024.00032
9. Adreani, L., Bellini, P., Fanfani, M., Nesi, P., Pantaleo, G.: Smart City digital twin framework for real-time multi-data integration and wide public distribution. IEEE Access. **12**, 76277–76303 (2024). https://doi.org/10.1109/ACCESS.2024.3406795
10. Fanfani, M., Palesi, L.A.I., Nesi, P.: Microservices' libraries enabling server-side business logic visual programming for digital twins. SoftwareX. **27**, 101805, ISSN 2352-7110 (2024). https://doi.org/10.1016/j.softx.2024.101805
11. Bellini, P., Bilotta, S., Palesi, A.L.I., Nesi, P., Pantaleo, G.: Vehicular traffic flow reconstruction analysis to mitigate scenarios with large city changes. IEEE Access. **10**, 131061–131075 (2022). https://doi.org/10.1109/ACCESS.2022.3229183
12. Bellini, P., Cenni, D., Palesi, L.A.I., Nesi, P., Pantaleo, G.: A deep learning approach for short term prediction of industrial plant working status. In: 2021 IEEE seventh international conference on big data computing service and applications (BigDataService), Oxford, United Kingdom, vol. 2021, pp. 9–16. https://doi.org/10.1109/BigDataService52369.2021.00007
13. Collini, E., Palesi, L.A.I., Nesi, P., et al.: Flexible thermal camera solution for Smart city people detection and counting. Multimed. Tools Appl. **83**, 20457–20485 (2024). https://doi.org/10.1007/s11042-023-16374-x
14. Murali, K., Ramakrishnan, K., Sandeep, S.: Efficient parking management for modern urban environments. In: 2024 3rd international conference on automation, computing and renewable systems (ICACRS), Pudukkottai, India, pp. 1456–1460 (2024). https://doi.org/10.1109/ICACRS62842.2024.10841537
15. Venkata Sudhakar, M., Anoora Reddy, A.V., Mounika, K., Sai Kumar, M.V., Bharani, T.: Development of smart parking management system. Mater. Today Proc. **80**(Part 3), 2794–2798, ISSN 2214-7853 (2023). https://doi.org/10.1016/j.matpr.2021.07.040
16. Janowski, A., Hüsrevoğlu, M., Renigier-Bilozor, M.: Sustainable parking space management using machine learning and swarm theory—the SPARK system. Appl. Sci. **14**(24), 12076 (2024). https://doi.org/10.3390/app142412076
17. Easypark, https://www.easypark.com/en-it/cities-and-operators
18. Smartparkingsystems, https://smartparkingsystems.com/en/smart-parking-system/
19. Bellini, P., Ipsaro Palesi, L.A., Nesi, P., et al.: Multi clustering recommendation system for fashion retail. Multimed. Tools Appl. **82**, 9989–10016 (2023). https://doi.org/10.1007/s11042-021-11837-5
20. Bellini, P., Bologna, D., Fanfani, M., Ipsaro Palesi, L.A., Nesi, P., Pantaleo, G.: Rapid prototyping & development life cycle for smart applications of internet of entities. In: 2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS), Toulouse, France, vol. 2023, pp. 142–151
21. Mondal, M.A., Rehena, Z., Janssen, M.: Smart parking management system with dynamic pricing. J. Amb. Intell. Smart Environ. **13**(6), 473–494 (2021). https://doi.org/10.3233/AIS-210615
22. Snap4city API, https://www.snap4city.org/drupal/node/20 (Accessed 04 Mar 2025)
23. Parking Comune Firenze, https://opendata.comune.fi.it/page_dataset_show?id=dati-in-tempo-reale-sui-posti-liberi-nei-parcheggi-di-firenze (Accessed 05 Mar 2025)
24. Bellini, P., Palesi, L.A.I., Giovannoni, A., Nesi, P.: Managing complexity of data models and performance in broker-based Internet/Web of Things architectures. Internet of Things. **23**, 100834, ISSN 2542-6605 (2023). https://doi.org/10.1016/j.iot.2023.100834

25. Bellini, P., Bologna, D., Nesi, P., Pantaleo, G.: A unified knowledge model for managing Smart City/IoT platform entities for multitenant scenarios. Smart Cities. **7**(5), 2339–2365 (2024). https://doi.org/10.3390/smartcities7050092
26. Adreani, L., Bellini, P., Fanfani, M., Nesi, P., Pantaleo, G.: Design and develop of a smart city digital twin with 3D representation and user Interface for what-if analysis. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications – ICCSA 2023 Workshops. ICCSA 2023 Lecture Notes in Computer Science, vol. 14111. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37126-4_34
27. Bilotta, S., Ipsaro Palesi, L.A., Nesi, P.: Predicting free parking slots via deep learning in short-mid terms explaining temporal impact of features. IEEE Access. **11**, 101678–101693 (2023)