SOPA: A Parking Solution

Rahul Singh, Mohd. Shahnawaz, Sonal Shriti, Nandita Pokhriyal, Kumar Anurag, Akshat Gupta, Vishal Jain
Dept. of Computer Science and Engineering
Sharda University, Greater Noida, Uttar Pradesh

Abstract-
The vehicle density has increased at an alarming rate and continues to do so. This is especially apparent during the peak hours of the day. This has made it very hard for vehicle owners to find a place to park their automobiles. This study aims to develop a mobile application by employing state-of-the-art technologies to tackle this widespread problem. The application uses data supplied by the users to give spot seekers a preferred location where they can park their vehicles. It also encourages users to supply more data by introducing in-app currency. This can help to combat problems like fuel wastage and save time as well.

Keywords- **Parking solution, mobile, spots, application**

I. INTRODUCTION

Parking is a day to day activity that characterizes life in cities. It is therefore the most common and necessary requirement of drivers to not only quickly but also efficiently search for parking spots [1]. Visiting new locations more often than not comes with an inherent parking problem. Something that inspires humor are jokes about holiday parking lot wars. But the amount of time spent looking for someplace to put your car at, on crowded city streets or mall lots has increasingly become a matter of grave concern. A 2011 IBM study claimed that almost 30 percent of a metropolis city's traffic can be attributed to people looking for parking slots, with a third of New York City drivers reporting that they spend on an average about 20 minutes searching on the road everyday [2]. Technological advancements in various domains have been a great help in increasing the Not Operating Income (NOI) for investors and have therefore aided in producing higher profitability. An example of which in the problem domain being discussed would be, sensors being used to detect the number of parking spaces that go unused [7]. These sensors are being utilized for a variety of purposes like determining the number of spaces that need to be maintained and thereby assisting in making marketing decisions [19]. An important point that goes mostly unnoticed is the fact that people are usually unaware of existing common/local parking spaces whenever they are visiting a new place, be it in the same city or a new city. Existing parking solutions are mostly targeted on solving the issues relating to managing parking space , smart parking, or parking management. The issue of locating a parking spot is highly important in the context of the parking management and has not been worked on enough or in the right way. Existing parking spot location solutions mostly suffer with the lack of parking spot data or lack of data quality and no incentive to add to the parking data. Our solution attempts to solve this problem by using crowdsourcing to collect parking data from users and a reliable rating system to maintain the quality of the data obtained from crowdsourcing. We also figured out that many users might have personal space which they would be willing to monetize by leasing the space for parking at certain times of the day. We have developed a solution to solve this problem by providing a way for users to lease their personal space for parking which any other user can see , book and use for the consent period by paying a fee.

Users are provided with incentive to add parking data to the system through a robust in-app token system that is used to reward users for contributing to the system and can be spent on searching for parking spots or other in-app services and features.

The remainder of this paper is organized as per the following scheme: Section 2 provides an analysis of some related as well as unrelated work that has been done in this field. Section 3 sheds light on the methodology used to carry out the implementation. The section also discusses, at length, the different modules that make up the application. Section 4 consolidates all the results from the implementation. Lastly, section 5 mentions the future scope of the implementation.

II. LITERATURE SURVEY

Urban cities becoming incredibly populated and hence congested teemed with the increase in vehicle ownership trends has led to the traffic crisis not only on roads but also in the public - private space management [20]. This of late has become a key issue with high economical impact. Today, cities are responsible for more than 75% of waste production, 80% of emission, and 75% of energy-utilization [3]. Taking the example of Europe, road transportation produces about 20% of the total CO2 emissions, out of which 40% of the emissions are generated by urban mobility [21]. Estimates state that the vehicles running and searching for unoccupied parking spaces cause about 30% of the daily traffic congestion in urban areas [4]. An effectively managed smart parking system would allow drivers to do away with the time they waste looking for ideal parking spots, an ideal situation which would not only help save time but would also consequently help in the decongestion of the roads thereby, improving the whole traffic situation, this would also ensure the optimum usage of parking spaces all the while playing a vital role in the domain revenue generation [25]. Even though the solution of infrastructure construction and public transport enhancement (e.g. buses and the legalization of the Uber service) might seem like the most obvious choice to deal with this problem it's not the most practical one, the use of technology on the other hand is another solution - a partial one to this bottleneck problem in the economy and the built environment[9]. Parking space demand is more of a dynamic entity as it changes from time to time, so if seen logically, providing real-time parking information (location and occupancy related) is not only difficult but also futile [22]. The origin of the smart apps meant to assist the parking situation can be traced back to around two decades back with systems like, "Guided Parking System" and "Dynamic Parking Guidance System" [10]. With the advent of Parking Guidance and Information (PGI) systems, improving parking efficiency has improved to a certain extent [11]. The effects of which are evident in the reduced noise levels, fuel consumption and exhaust emissions (due to a decrease in the vehicle count at certain segments of the traffic and also due to shorter travel times). Farooqi et. al. [18] in his work on UParking provides a complete solution, with ANPR cameras capturing vehicle plate numbers at the parking location and matching the obtained information against their database before providing the vehicle entry into the lot. Kurek et. al. [13] goes on to describe various typologies found in contemporary solutions, contemporary to ours is the typology based on Parking Guidance and Information (PGI) systems' smart applications section where it has been described to have made the use of app sourced data along with GPS as a possible answer to providing navigational parking aid. The PGI systems typically consist of four constituents, monitoring parking lots, disseminating the information thus collected, communication technology, and the control systems [15]. These systems use either cameras or sensors to monitor the lot for possible occupancy. In Japan, the updating of the PGI information is done via a team of what they've termed as "walking surveyors" for entering data into GPS navigational systems (mounted on vehicles), that would show car park vacancies [23]. An alternative solution to Japan's is using a bunch of vehicles equipped with a variety of sensors to capture the street scenes and situation for transforming into (Geographic Information System) GIS data [24]. The sensor based techniques can be further classified into the off-road and on-road, depending on the kind of sensors used. The knowledge body in this arena is ever expanding with enormous deployments since 1996[16].

Anand et. al. [8] has included IOT to carry out the implementation. The application requires the user to register themselves in the app first in order to use the services. A registration card is then given to the drivers which is issued on a one-time basis, and the data of the driver is stored in the database in Raspberry Pi. Teong Ang et. al. [17] has proposed in his work a somewhat upgraded solution to the one proposed by Anand et. al [8], in that his proposed system iSCAPS uses Near Field Communication (NFC) function of smart phones as a stand-in for the aforementioned registration cards. An arduino microcontroller acts as the brain of this system which also provides an additional functionality of searching for one's vehicle in case one forgets the spot they had parked their vehicle in.

There exist several ways in which these smart parking systems can be designed employing the array of technologies that are on the rise. Some of which are sensor based solutions such as magnetic and acoustic sensors (on-road), ultrasounds, RFIDs, smart cameras, etc. A more advanced solution for the same is crowdsourcing, which is based on using ultrasonic sensors fitted into vehicles, for detecting empty slots near a parked vehicle followed by information dissemination to drivers searching for these slots. “There are also apps that give information on on-street parking, recommending the best zones to search for unoccupied parking spots, or giving the opportunity to announce when a user is about to move a vehicle, leaving free space, and in some cases, allowing reservations for these new free spots. Each of the above mentioned functionalities can use the data that has been obtained automatically via cell phones (crowdsensing) or provided manually by the app users (crowdsourcing). While off the street parking information has the potential to help users reduce the price to pay or to select the closest available space at their destination, on the street parking information usually has as its main goal the reduction of multiple cars chasing a single space”[5]. Parking more often than not is considered an arduous task, one that requires the driver to be in possession of a certain level of expertise, in order to be able to safely and properly maneuver the vehicle into position. Perpendicular parking, among the many other methods, is considered the most complex, especially taking drivers with disabilities into perspective. Gamal et. al.[12] has, in his work, tackled this problem of perpendicular parking with the help of Deep Convolutional Neural Networks (CNNs). The neural network has been trained to imitate the Automatic Parking Assist Systems (APAS). Another work in this domain by F. Alshehri et al. [14], deals with the problem of “wrong parking”- vehicles being parked out of angles or at odd angles, by using a simple method that monitors the cars which are parked in a parking lot via two arduinos, one fixed and one mobile, along with a ultrasonic sensor, PIR motion sensors and a Nexion display attached to the fixed arduino to display the results. While the solution is effectively area/ park specific it does provide a working idea for a human proctor to act on, by color coded information on the Nexion screen of the issues the system has been made to deal with.

In [6] the authors have suggested something called pocketsourcing for the automatic detection of available parking slots by making use of mobile phone sensors. The experimental results have been claimed to be 94% correctly predicted. A smart parking application called ParkTAG uses an implementation where an automatic detection algorithm is used, this type of strategy is currently being used by various smart apps in European countries. The main barrier faced with an approach of this sort is to persuade the users to let the application keep continuing its operation in the background [26]. Another issue faced at the beginning stages of deployment would be insufficient data for the algorithm to work on due to lack of users, this is called a cold start problem which was solved in the aforementioned Japanese approach of using surveyors. Other works in the area contemporary to ours, focus on the overall enhancement of the working task by taking something similar to a cost function whose value relies on the distance from the destination and the money charged for the parking space.

III. METHODOLOGY

A. REQUIREMENTS

The requirements consist of a set of features that we would like to implement in our system. We are aiming for a mobile app since that is the simpler platform commuters use. We decided to go with React Native framework for the mobile app since it is cross platform, i.e, we can use it to create mobile applications for both iOS and Android systems which combined accounts for the majority of the users. Next, we require a database and cloud storage services to store our growing parking location data , any related metadata or the user information which should be accessible to all users. We decided to go with mongoDB since it is a scalable database and is easy to use. For the storage service we opted for AWS S3 since we were already aiming for Amazon Web Services for the cloud since it has a lot of affordable features and a strong community with documentations for every service. Next, we finalized our backend framework that is supposed to be running the application on cloud. We decided to go with FastAPI since it is a lightweight framework that is easy to use and allows easier scalability and documentation tools. We took the microservices approach for the backend instead of the traditional Monolith approach since it is easier to scale and allows for easier testing and can be smartly used to increase the cost effectiveness and uptime of the application. Next comes the payment services that are supposed to be implemented for private spot lease and parking management. We decided to go with Razorpay since it is a popular payment service that is easy to use and has a lot of features that are useful for our application.

B. SYSTEM ANALYSIS

Some well known applications and their features are given in Table 1.

S No.	Features	Parking Rhino	Best Parking (by Arrive)	Park Plus	SOPA (our solution)
1	Review/ Report	✓	✓	✓	✓
2	Location Search	✓	✓	✓	✓
3	Location wide slot display	✓	✓	✓	✓
4	Free Parking options	✓			✓
5	Pay and park options	✓	✓	✓	✓
6	Premium				✓
7	Easy Payments	✓	✓	✓	✓
8	Refund				✓
9	In-app Currency				✓
10	Users can upload potential Parking locations		✓	✓	✓
11	Spot Reservation		✓		✓ (PrivateSpots)
12	Distinct Markers	✓			✓
13	Real-Time Navigation to Parking Location	✓	✓	✓	✓
14	Detailed location information	✓	✓	✓	✓
15	Lease out a spot				✓

Table1. A comparison between some existing solutions and our solution

1) Class Diagram

The UML Class Diagram in Fig1. describes the structure of the proposed system and defines the classes, operations and attributes present in the systems. It also defines the relationships between the various objects.

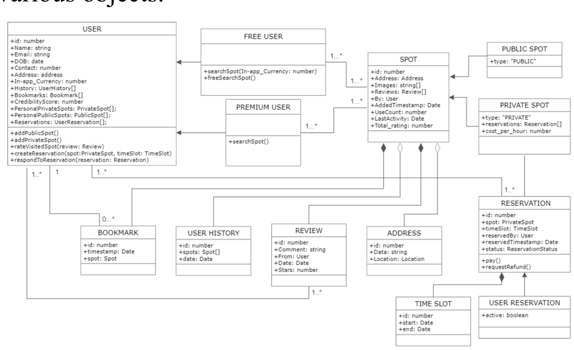

Fig1. UML Diagram

2) Use Case

The Use Case Diagram in Fig2. summarizes all the interactions that the user will have with the proposed system.

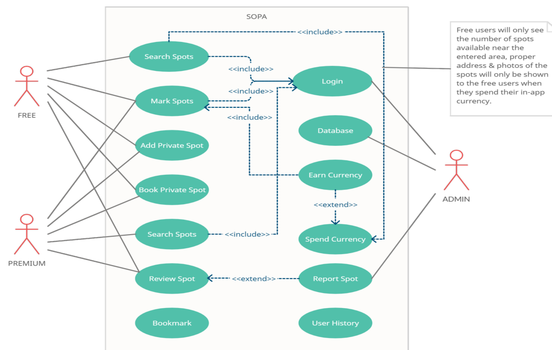

Fig2. Use Case Diagram

3) System Architecture

Our current architecture has three internal services namely "Common Service", for all common user operations, "Premium Service" for operations related to Premium Users, and a "AAA service", which fulfills the purpose of Authentication, Authorization and Accounting operations. We have also featured a review system architecture wherein the reviews are sent to a notification service (SNS) which adds it to two queues, one for queuing reviews for filtering and other as a control queue which limits the horizontal scaling of the Fargate instances. The review filtering systems can be deployed to Fargate which are cheap and readily available cloud services.

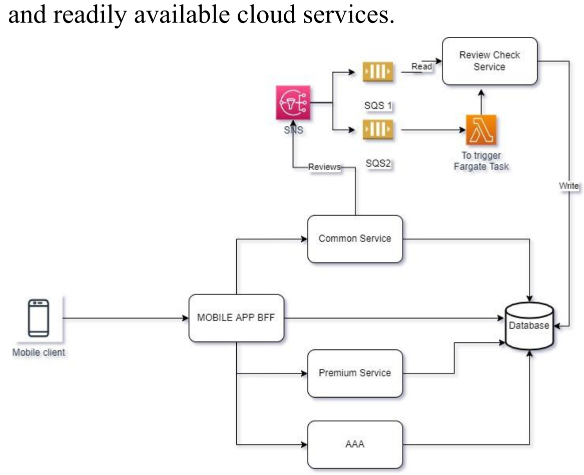

Fig3. System Architecture

C. IMPLEMENTATION

The review verification system is currently a very simple one with room for enhancement in the future since our architecture allows for a lot of flexibility and capabilities for the system. Initially we will be implementing a basic word filtering system that will be used to filter out the reviews that use blocked words. Our payment systems had to be carefully planned so as to allow room for fairly solving any disputes between the two parties. We decided to implement a payment system that is based on the Razorpay payment service. Initially on receiving a payment we will be temporarily holding the amount within our system for an agreeable amount of time during which the paying party can raise any disputes which then will be to be resolved with the receiver party under our mediation and only upon the resolution of the dispute will the amount be released to the paying or receiving party. Our parking location recommendations will work on the basis of the nearest and highest rated locations based on the number and recency of reviews, recent reviews being given a higher weightage allowing for any edge cases where a previous popular parking spot might not exist anymore or have gone down in quality.

1) Authentication

User sends their mobile number and any other related metadata to the Mobile BFF which validates and then further passes them on to the Authentication Authorization Server which logs in or signs up user depending on the request by generating an OTP through Twilio service , an OTP_ID and a auth token when needed.

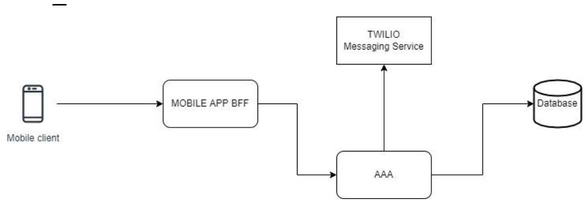

Fig4. Authentication System Design

2) Add a Parking Spot

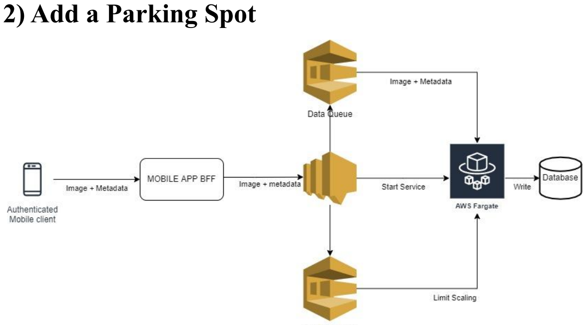

Fig5. Add a Spot Architecture

Parking Spot Data

- Images - Array of images. At least one image is required.
- Name - Name of the parking spot.
- Description - Description of the parking spot.
- Address - Geo-coordinates of the parking spot.
- Type - Type of the parking spot. Can be Public or Private for the user.
- Price - If user selects a **private** type for their spot, they will have to specify a price per hour for renting the spot.
- User - User ID of the user who created the parking spot.

Authenticated Mobile Client sends a parking spot image along with relevant location and description metadata to the Mobile BFF which then passes it on to SNS (Simple Notification Service). Notification Service then adds this data to two queues:

- Control Queue : This queue is used to limit the horizontal scaling for the fargate service.
- Data Queue : This queue is used to queue the data that is needed by the fargate service.

After validating the image and metadata, the Fargate Service then appends the data to the database and deletes it from the Data Queue.

It also checks for any other data available in the queue before ending the service.

Upon validation of a parking spot, the user receives in-app currency to their account.

3) Search a Parking Spot

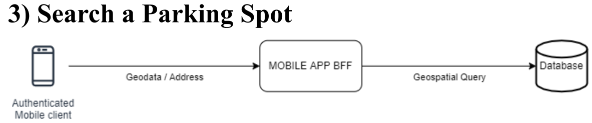

Fig6. Search a Parking Spot Architecture

Authenticated Mobile Client sends geo-coordinates or an address to the Mobile BFF which incase of an address uses an external service to convert that to geo-coordinates runs a query on the database for nearby locations and returns the results.

Free/Regular Users

Free users will not receive an exact location but only a proximity for the parking spots. They can then choose to spend their in-app currency to view the exact location.

4) Rent a Parking Spot

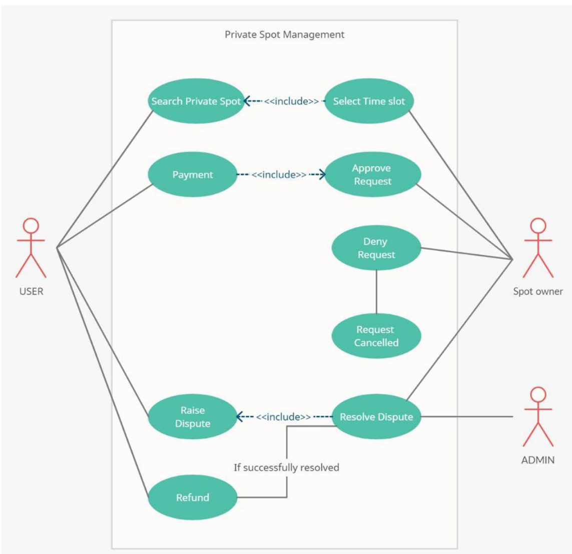

Fig7. Use Case Diagram for Renting a Spot

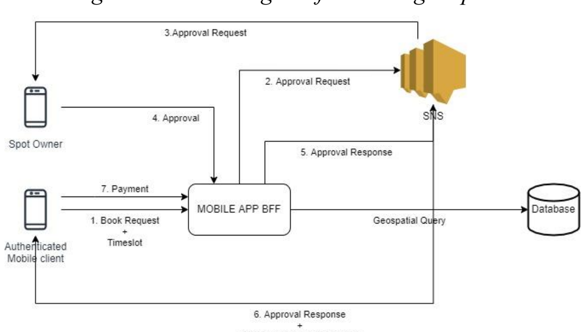

Fig8. Rent a Spot

An authenticated mobile client makes a request to the Mobile BFF for renting a private spot. BFF then checks if the spot is available and if it is, it sends an approval request to the Parking Spot Owner through SNS. Upon acceptance of the request, the mobile client is notified and presented with a payment channel to complete payment to complete the process. Upon the completion of payment, the owner is notified and the spot is marked as occupied for the time slot. The payment is then transferred to the Spot Owner after a cooldown time incase of no disputes.

IV. RESULTS AND CONCLUSIONS

The paper presents a smart, crowdsourced parking location search solution based on modern cloud technologies. The SOPA system combines a mobile application that offers a variety of features and services, collecting and sharing parking data and incentifying users to do so that makes it a self-sustaining solution. The mobile application is supported by an interconnected microservice architecture on the cloud. This system proposes an all round solution to finding a relevant and safe parking spot. The system will also be extended to cover additional new features. It will be tested practically in various situations.

V. FUTURE SCOPE

The future scope of this system hopes to improve spot image verification once the system gathers a bulk of public spot images in the database from crowdsourcing. This can help to build models to better predict if the spot image is relevant and help to add a scoring algorithm for the in-app currency. Better language checks can also be implemented and profanity filters for spot reviews.

REFERENCES

[1] Kim, Soyeon, et al. "Smart Parking with Learning-aided User Activity Sensing Based on Edge Computing." *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021.

[2] Kozłowski, Wojciech, and Kacper Suwar. "Smart City: Definitions, Dimensions, and Initiatives." *European Research Studies* 24 (2021): 509-520.

[3] Lam, Patrick TI, and Wenjing Yang. "Application of technology to car parking facilities in Asian smart cities." *Journal of Facilities Management* (2019).

[4] Ji, Zhanlin, et al. "A cloud-based intelligent

car parking services for smart cities." *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*. IEEE, 2014.

[5] Ji, Zhanlin, et al. "A cloud-based car parking middleware for IoT-based smart cities: Design and implementation." *Sensors* 14.12 (2014): 22372-22393.

[6] Enríquez, Fernando, et al. "Existing approaches to smart parking: An overview." *International Conference on Smart Cities*. Springer, Cham, 2017.

[7] Kan, Zihan, Man Sing Wong, and Rui Zhu. "Understanding space-time patterns of vehicular emission flows in urban areas using geospatial technique." *Computers, Environment and Urban Systems* 79 (2020): 101399.

[8] Anand, Abhijeet, et al. "Smart Parking System (S-Park)–A Novel Application to Provide Real-Time Parking Solution." *2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)*. IEEE, 2020.

[9] Cheshire, David. *Building revolutions: Applying the circular economy to the built environment*. Routledge, 2019.

[10] Milosavljevic, N., & Simicevic, J.. Parking guidance and information system. *Sustainable Parking Management*, 2019.

[11] Di Martino, Sergio, Vincenzo Norman Vitale, and Fabian Bock. "Comparing different on-street parking information for parking guidance and information systems." *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019.

[12] Gamal, Omar, et al. "Assistive parking systems knowledge transfer to end-to-end deep learning for autonomous parking." *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*. IEEE, 2020.

[13] Kurek, Agata. "Methods of Parking Measurements—Research of Parking Characteristics in Paid Parking Zones with Dynamic Parking Information." *Research Methods in Modern Urban Transportation Systems and Networks*. Springer, Cham, 2021. 185-193.

[14] Alshehri, Faris, et al. "Smart Parking System for Monitoring Cars and Wrong Parking." *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019.

[15] Nie, Yan, et al. "Crowd-parking: A new idea of parking guidance based on crowdsourcing of parking location information from automobiles." *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1. IEEE, 2019.

[16] Jung, Andrea, Paul Schwarzbach, and Oliver Michler. "Future Parking Applications: Wireless Sensor Network Positioning for Highly Automated in-House Parking." *ICINCO*. 2020.

[17] Ang, Jin Teong, et al. "iSCAPS-Innovative Smart Car Park System integrated with NFC technology and e-Valet function." *2013 World Congress on Computer and Information Technology (WCCIT)*. IEEE, 2013.

[18] Farooqi, Norah, et al. "UParking: Developing a smart parking management system using the internet of things." *2019 Sixth HCT Information Technology Trends (ITT)*. IEEE, 2019.

[19] Kanan, Riad, and Houssam Arbess. "An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing." *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020.

[20] Tian, Zhao, et al. "Determination of key nodes in urban road traffic network." *Proceeding of the 11th World Congress on Intelligent Control and Automation*. IEEE, 2014.

[21] Hallik, Lea, et al. "Reflectance measurements at climate change experiment sites in Europe." *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*. IEEE, 2013.

[22] Tingting, Tong, and Zhou Jin. "Space-Time Model of Urban Parking Charge." *2009 Second International Conference on Intelligent Computation Technology and Automation*. Vol. 2. IEEE, 2009.

[23] Caixia, Liu, and Li Daosheng. "Comparative Review of Logistics Park Planning among China, Japan and Germany: response to territorial development in China." *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021.

[24] Zhang, Zusheng, et al. "Collaborative sensing-based parking tracking system with wireless magnetic sensor network." *IEEE Sensors Journal* 20.9 (2020): 4859-4867.

[25] Assim, Marwa, and Alauddin Al-Omary. "A survey of IoT-based smart parking systems in smart cities." (2021): 35-38.

[26] Chen, Mingsong, Runze Fang, and Lei Peng. "Exploration of Parking Guidance based on Vehicle Crowdsourcing." *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2020.