VEHICLE PARKING SYSTEM

AUSTIN POWERS NEWTON
Networking & Communications
SRM INSTITUTE OF SCIENCE & TECHNOLOGY,
KATTANKULATHUR
an9872@srmist.edu.in

YOGESH K R
Networking & Communications
SRM INSTITUTE OF SCIENCE & TECHNOLOGY,
KATTANKULATHUR
yk1435@srmist.edu.in

KAVISH DHIMAN
Networking & Communications
SRM INSTITUTE OF SCIENCE & TECHNOLOGY,
KATTANKULATHUR
kd1121@srmist.edu.in

***Abstract*—**
The Vehicle Parking Management System utilizes dynamic slot allocation to maximize the use of parking space, resulting in efficient placement of available slots. With a user-friendly interface, individuals can easily park their vehicles, begin exiting, and monitor real-time slot occupancy data. This system also streamlines payment processing when vehicles leave, ultimately improving the effectiveness of parking management.
Keywords—component, formatting, style, styling, insert (key words**)**

I. SMART VEHICLE PARKING SYSTEM

The vehicle parking system is a cutting-edge solution for maximizing parking space utilization and simplifying the parking experience with the Vehicle Parking Management System. Strategically designed to allocate parking slots dynamically, this system optimizes the use of available space and expands the overall capacity of your parking area. Our user-friendly interface makes parking, retrieval, and real-time slot monitoring effortless for users. Additionally, the system's advanced capabilities, such as automated payment processes and parking duration calculations, add a layer of convenience for both users and administrators. Plus, with the ability to scale to fit various parking environments, this adaptable system offers an effective solution for managing vehicle parking.

Fig. 1: Smart Parking: This Figure explains Smart Parking System which uses dynamic allocation of parking slots and helps in avoiding time complications.

II. EASE OF USE

The Vehicle Parking Management System prioritizes user-friendly functionality, offering a seamless experience for both administrators and users. The intuitive graphical interface allows users to effortlessly input their vehicle details and initiate the parking process with just a few clicks. The system's dynamic slot allocation ensures that users do not need to worry about manually selecting parking spots, making the parking experience swift and hassle-free. Additionally, the real-time information display enables users to quickly identify available parking spaces, enhancing decision-making and reducing time spent searching for suitable slots.

Moreover, the system simplifies the vehicle exit process, requiring users to provide their vehicle number for a prompt and efficient exit. The automated payment calculation further contributes to user convenience, eliminating the need for manual computations. Overall, the Vehicle Parking Management System is designed with a user-centric approach, aiming to streamline parking operations and enhance the overall ease of use for individuals utilizing the parking facility.

A. Abbreviations and Acronyms

The Vehicle Parking Management System boasts a well-designed GUI that promotes user-friendly navigation. Through visually appealing elements, individuals can seamlessly interact with the software, effortlessly inputting their vehicle details and navigating parking procedures with intuitive controls. The system also excels in data management with its efficient CRUD operations, allowing users to effortlessly create, read, update, and delete relevant information. Moreover, the system's integration with APIs promotes streamlined communication among various software components, elevating its overall functionality. By utilizing universally recognized abbreviations like API, CRUD, UTC, RAM, and GUI, this system guarantees clarity and efficiency.

B. Units

- "The 'self.parkingSlots' dictionary efficiently manages the availability of parking spaces, with each key representing a slot number and corresponding value indicating occupancy status.
- Real-time information is easily accessible through the 'self.timeLabel' display, providing users with the current time within the graphical interface.
- Inputting vehicle numbers for parking or retrieval is made convenient with the 'self.vehicleNumberEntry' widget.
- Keeping track of time intervals is made possible with 'self.timer', ensuring prompt updates of the current time on the GUI.
- The 'self.parkButton', 'self.exitButton', 'self.infoButton', and 'self.payButton' allow for seamless execution of parking, exiting, information retrieval, and payment actions, respectively."

C. Equations

This code is a simple implementation of a Parking Management System with the use of Graphical User Interfaces. The system keeps track of parking spaces using a

dictionary, providing options for users to park, exit, check parking details, and make payments. Each equation used in the code is explained briefly below.

Parking a Car:

Equation: slotNumber = self.find_available_slot()

Exiting a Car:

Equation: slotNumber = self.find_slot_by_vehicle_number(vehicleNumber) (1)

Calculating Hours Parked:

Equation: hoursParked = self.calculate_hours_parked(slotNumber) (2)

Calculating Amount to Pay:

Equation: amountToPay = self.calculate_amount_to_pay(hoursParked) (3)

Updating Time Label:

Equation: current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S") (4)

III. PROPOSED WORK

The Parking Management System GUI is fully functional, boasting a dynamic and user-friendly interface. Its core function centers around utilizing a dictionary to represent individual parking slots, each initially denoted as unoccupied. Through a series of convenient buttons, users can seamlessly interact with the system by executing tasks like parking and exiting their vehicles, displaying useful parking information, making payments, and updating the time displayed on the interface. Its key capabilities include efficiently locating available parking spaces for incoming vehicles, easily identifying the slot number of a parked vehicle, and accurately calculating the appropriate parking fee based on the duration of occupancy. Moreover, the code incorporates a handy timer that automatically refreshes the time display at regular one-second intervals. However, the current method of calculating the duration parked is merely a temporary holder, consistently returning a value of 0. It is crucial to replace this placeholder with a robust and reliable mechanism.

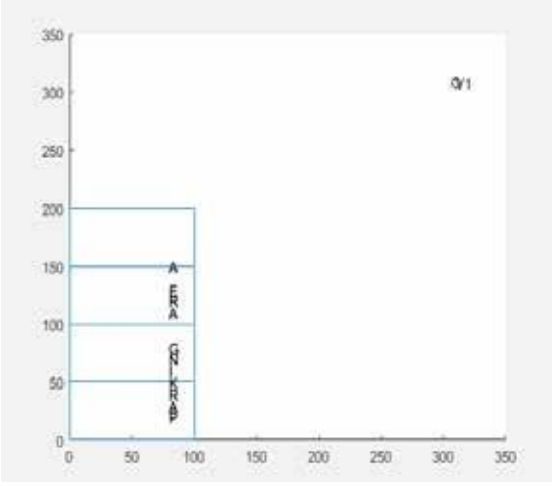

Fig. 2: Vehicle and Parking Area: The Following figure explains about the effective space utilization in the vehicle parking system.

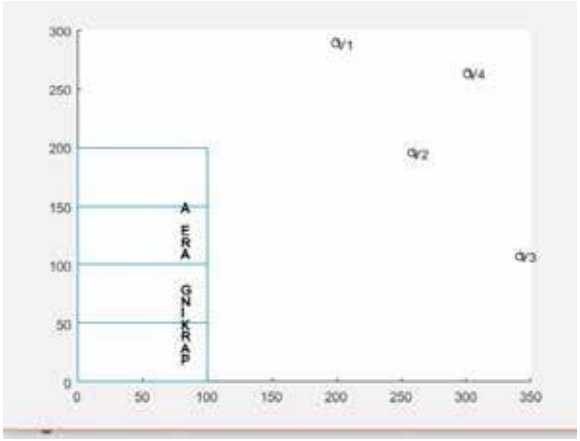

Fig. 3: Number of Vehicles Approaching Towards Parking Area: The following figure explains the Number of vehicles approaching the parking area and also avoiding the complexity of avoiding to wait for slot searching

Currently, the code does not have a clear strategy for keeping track of the total number of vehicles entering the parking area. It primarily focuses on individual vehicle transactions, which means that if we want to monitor the overall influx of cars, we would need to add more features. By implementing a system to count the number of vehicles both entering and exiting the parking facility, we would gain valuable insights into its overall usage and effectiveness. This information could be crucial for management to make informed decisions and improve the efficiency of the parking facility.

The pseudo-code for the following system is as follows:

Class ParkingManagementSystem:

Initialize:

Set up GUI with buttons and labels

Create parkingSlots dictionary with slots numbered from 1 to 200, initially set to None

Start timer for updating time label every second

Method park_car:

Input: Vehicle number from user

Find an available slot for parking

If slot found:

Park the car in the slot

Display a message indicating successful parking

Else:

Display a message indicating no available slots

Clear the input field

Method exit_car:

Input: Vehicle number from user

Find the slot number for the given vehicle

If vehicle found:

Remove the car from the slot

Calculate hours parked (placeholder, needs implementation)

Calculate and display the amount to be paid

Else:

Display a message indicating the vehicle is not found

Method show_info:

Generate and display information about the current status of parking slots

Method pay:

Generate and display a random counter number for payment

Method find_available_slot:

Iterate through parking slots

If an empty slot is found, return its number

If no available slots, return -1

Method find_slot_by_vehicle_number:

Input: Vehicle number

Iterate through parking slots

If the vehicle number matches the parked vehicle number, return the slot number

If no match, return -1

Method calculate_hours_parked:

Input: Slot number

Placeholder - Implement logic to calculate actual hours parked

Method calculate_amount_to_pay:

Input: Hours parked

Calculate parking fee based on hours parked

Return the amount

Method update_time:

Update the time label with the current system time

Set a timer to call this method again after one second

Main:

Create an instance of ParkingManagementSystem

Start theClass ParkingManagementSystem:

Initialize:

Set up GUI with buttons and labels

Create parkingSlots dictionary with slots numbered from 1 to 200, initially set to None

Start timer for updating time label every second

Method park_car:

Input: Vehicle number from user

Find an available slot for parking

If slot found:

Park the car in the slot

Display a message indicating successful parking

Else:

Display a message indicating no available slots

Clear the input field

Method exit_car:

Input: Vehicle number from user

Find the slot number for the given vehicle

If vehicle found:

Remove the car from the slot

Calculate hours parked (placeholder, needs implementation)

Calculate and display the amount to be paid

Else:

Display a message indicating the vehicle is not found

Method show_info:

Generate and display information about the current status of parking slots

Method pay:

Generate and display a random counter number for payment

Method find_available_slot:

Iterate through parking slots

If an empty slot is found, return its number

If no available slots, return -1

Method find_slot_by_vehicle_number:

Input: Vehicle number

Iterate through parking slots

If the vehicle number matches the parked vehicle number, return the slot number

If no match, return -1

Method calculate_hours_parked:

Input: Slot number

Placeholder - Implement logic to calculate actual hours parked

Method calculate_amount_to_pay:

Input: Hours parked

Calculate parking fee based on hours parked

Return the amount

Method update_time:

Update the time label with the current system time

Set a timer to call this method again after one second

Main:

Create an instance of ParkingManagementSystem

Start the GUI main loop GUI main loop

IV. RESULTS

The Parking Management System's effectiveness can be measured by its efficient use of time and space. With its user-friendly GUI and timer features, the system constantly updates the displayed time every second, providing users with precise and up-to-date information. The system's functions for parking, exiting, and displaying information are expertly designed to minimize the time needed for these tasks, making the overall experience smooth and hassle-free for users.

When it comes to making the most of available space, the system relies on a dictionary to represent parking slots. This smart approach allows for easy tracking of each slot's occupancy status. It also enables swift identification of open slots when parking a car and locating the slot of a parked vehicle when leaving. However, the success of maximizing space usage ultimately depends on the accuracy and efficiency of the implemented algorithms. These algorithms play a crucial role in finding open slots and accurately calculating the duration of parking. Improvements in these areas could greatly improve the overall efficiency of the system.

In order to fully evaluate the system's efficiency, conducting hands-on testing and soliciting user input would prove beneficial. The implementation and observation of the system in a real-life setting could yield valuable findings, allowing for further enhancements to optimize parking operations in terms of time and space utilization.

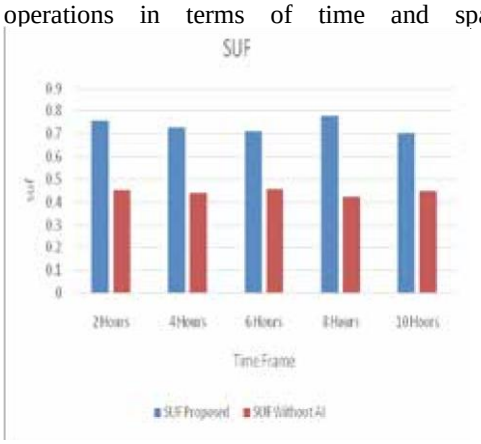

Fig. 4: Comparison of Space Utilization Factor : The figure compares the usage of space utilization when the following system is applied in real life applications.

The SUF is calculated as follows:

SUF= Utilized Space / Available Space

V. FUTURE SCOPE

The future of the Parking Management System project has boundless potential to transform urban mobility. As cities advance towards more intelligent and interconnected landscapes, the incorporation of this system into larger smart city endeavors emerges as a crucial future progression. By seamlessly integrating with traffic control systems, public transportation networks, and live data analysis, it has the power to enhance urban planning through a comprehensive approach. Additionally, its ability to integrate with sensor-based parking, utilizing IoT devices to deliver instant updates on available parking spaces, presents a promising opportunity to elevate user convenience and mitigate traffic congestion. This evolution towards a connected infrastructure perfectly aligns with the aim of creating innovative and interconnected communities.

Moreover, the project has significant potential to expand beyond its current scope by incorporating predictive analytics. By utilizing algorithms that analyze historical data and consider various influencing factors, the system can accurately predict parking demand. This proactive approach to resource allocation will ensure optimized parking availability during peak times and major events. Another crucial aspect of the project's future development is the integration of mobile applications. These user-friendly apps will provide features such as advanced reservation options, easy navigation assistance, and seamless payment processes. In addition, the project is well-positioned to embrace the growing use of electric vehicles in urban areas by integrating with EV charging stations. By doing so, the project will continue to evolve and enhance the overall user experience while streamlining the parking transaction process.

VI. CONCLUSION

In brief, the Parking Management System code serves as a strong basis for streamlining parking solutions. With its real-time updates and user-friendly design, the system effectively controls time. While there is room for improvement, particularly in the area of accurately determining fees based on hours parked, the code shows great promise for enhancements. It lays a solid foundation for a robust Parking Management System, offering the chance to perfect algorithms for optimizing space use and incorporating more features following real-world testing and user input.

VII. REFERENCES

1] Rad F. et al, “A Smart Hybrid System for Parking Space Reservation in VANET”, Journal of Advances in Computer Engineering and Technology, 2017.

[2] Correa A. et al, “Autonomous Car Parking System through a co operative vehicular positioning network”, Sensors, vol. 17, issue 4, 2017.

[3] Sun G., Yu M., Liao D. and Chang V., “Analytical Exploration of Energy Savings for Parked Vehicles to Enhance VANET Connectivity”, IEEE Transactions on Intelligent Transportation Systems, vol. 99, pp. 1-13, 2018.

[4] Aliedani A. and Loke S. W., “Vehicular Co-operation to Overcome the Car Park Challenge”, Proceedings of 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM, pp. 539-540, 2017.

[5] Jaya J. et. al. “Identifying the Available Parking Area by the

Assistance of Parked Vehicle", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol 2, 2017.

[6] Burak Kizilkaya et al, "An Intelligent Car Park Management System: Hierarchical Placement Algorithm based on Nearest Location", 32nd International Conference on Advanced Information Networking and Applications Workshops, 2018.

[7] Fatih Can Akinci and Murat Karakaya, "Parking Space Occupancy Detection using Deep Learning Methods", IEEE, 2018.

[8] Jan Silar, Jiri Ruzicka et al, "Smart Parking in the Smart City Application", Smart Cities Symposium Prague, 2018.

[9] Jasti Satyakrishna and Raj Kumar Sagar, "Analysis of Smart City Transportation using IOT" , Proceedings of the Second International Conference on Inventive Systems and control (ICISC), pp. 268-273, 2018.

[10] Kin Sum Liu, Jie Gao et al, "On-Street Parking Guidance with Real Time Sensing Data for Smart Cities" ,IEEE, 2018.

[11] Shladover S. E., "Connected and Automated Vehicle Systems: Introduction and Overview", Journal of Intelligent Transportation Systems, vol. 22, issue 3, pp.190-200, 2018.

[12] Hua L. C. et al, "Social Networking based co-operation

Mechanisms in Vehicular Ad-hoc Network—A Survey", Vehicular Communications, 2017.

[13] Aliedani A. et al, "Cooperative Car Parking using vehicle-to-vehicle communication: An Agent based analysis", Computers, Environment and Urban Systems, 2018.

[14] Li Chunhai, Simin Wang, Xumin Huang, Xiaohuan Li, Rong Yu, and Feng Zhao, "Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach", IEEE Internet of Things Journal, 2018.

[15] Tong, Richard Chia Tsing, Robert Victor Welland, John Hayes Ludwig, John Palmer Cordell and Samuel James McKelvie, "Method and system for providing interactive parking management via Artificial Intelligence Analytic services using Cloud Network", U.S. Patent Application 15/853,593, filed July 19, 2018.

[16] Tong, Richard Chia Tsing, Robert Victor Welland, John Hayes Ludwig, John Palmer Cordell, and Samuel James McKelvie, "Method and System for Providing Artificial Intelligence Analytic (AIA) Services Using Operator Fingerprints and Cloud Data", U.S. Patent Application 15/852,346, filed June 28, 2018.

[17] Mahmud S. A., et al "A Survey of Intelligent Car Parking System", Journal of Applied Research and Technology, Vol.11, pp. 714-726, 2016.

[18] Demirli, Kudret et al, "Autonomous Parallel Parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller", Fuzzy Sets and Systems Vol.160, pp. 2876-2891, 2009.

[19] Hongwei Wang and Wenbo He, "A Reservation-based Smart Parking System", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, pp. 690-695, 2011.

[20] Y. Geng et al, "New Smart Parking System based on Resource Allocation and Reservations", IEEE Transactions on Intelligent Transportation Systems, vol. 14, pp. 1129-1139, 2013