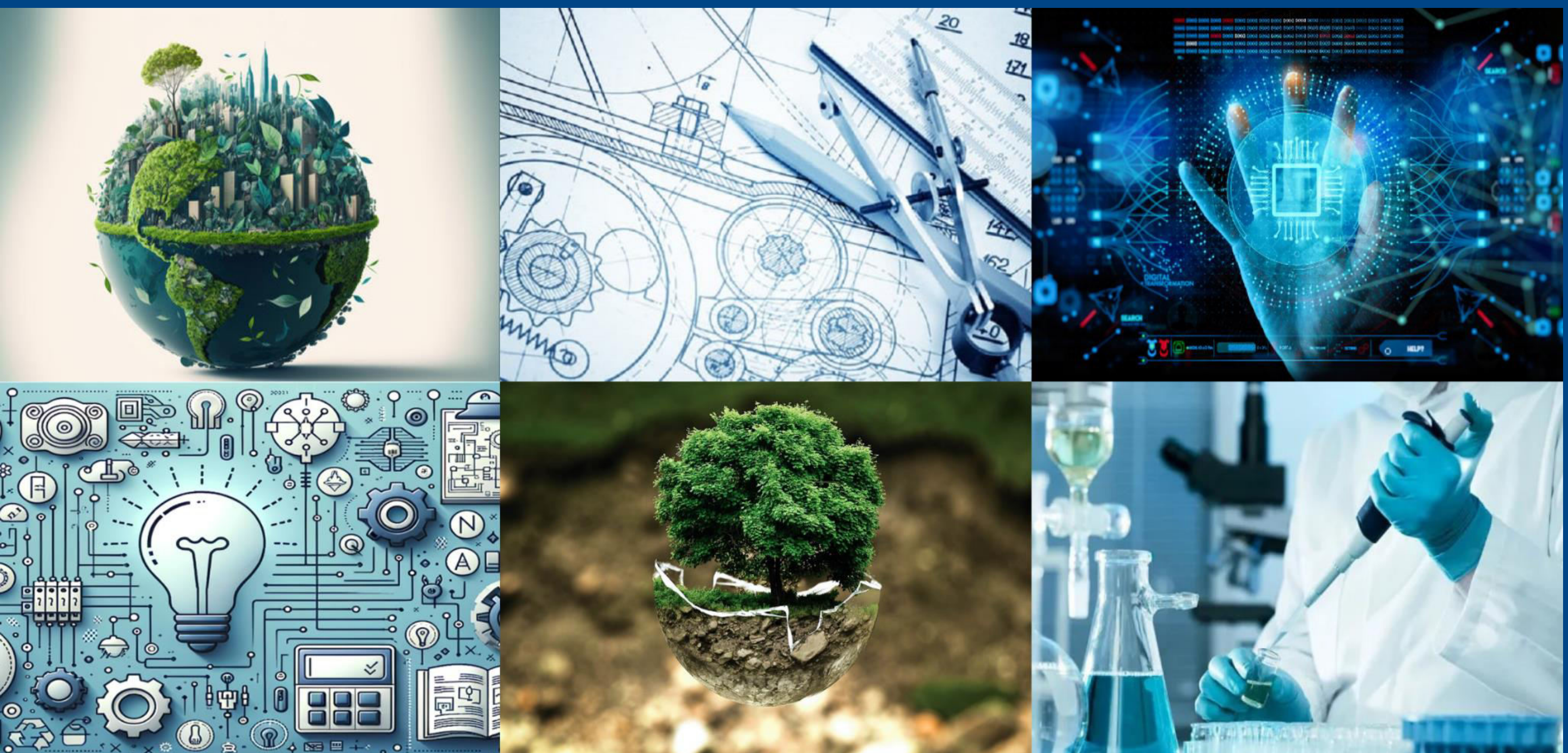
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 6, June 2025

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806161

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Optimized Smart Parking System

Aishwarya Auti[*1], Chitra Ragit[*2], Rutuja Aware[*3], Shradha Arsewar[*4]

Dept. of Computer Engineering, Sinhgad College of Engineering, India [*1, 2, 3, 4]

ABSTRACT: The Optimized Smart Parking System addresses the complexities of urban parking through a mobile-first, real-time Android application. Users can conveniently locate, reserve, and pay for parking slots via the app, eliminating the need for on-site searching. The system utilizes Google Firestore for seamless real-time data updates and Firebase Cloud Messaging to deliver intelligent notifications. A dynamic pricing mechanism adjusts parking rates based on demand and time, promoting efficient space allocation. Payment processing is securely handled through Razorpay, ensuring quick and reliable booking confirmations. Instead of physical sensors, the system relies on GPS technology and cloud-native infrastructure, reducing hardware dependency. System stability and performance were validated using tools like Selenium and JUnit, while development followed the Agile methodology to support continuous feature enhancements. User acceptance testing reflected higher satisfaction levels and increased operational efficiency, with search times reduced by up to 50%, contributing to lower traffic congestion. Looking ahead, planned upgrades include the incorporation of predictive analytics and deeper integration with broader smart city ecosystems.

KEYWORDS: Smart Parking System, Real-Time Slot Booking, Android Application, Firebase, Google Firestore, Razorpay Integration.

I. INTRODUCTION

Urbanization has significantly increased vehicle ownership, intensifying parking challenges in cities due to limited infrastructure and growing mobility demands. Drivers often face difficulties in finding available parking spots, leading to traffic congestion, unnecessary fuel consumption, and elevated emissions. Traditional parking systems, typically dependent on manual oversight and fixed pricing, are inefficient and lack real-time adaptability. In response, smart parking solutions have emerged, integrating IoT devices, mobile applications, and cloud technologies to address these issues. However, many of these rely on expensive sensor networks and complex hardware installations, making them impractical for widespread deployment.

This paper introduces an Optimized Smart Parking System—a lightweight, scalable Android application that empowers users to search, reserve, and pay for parking slots in real time. It leverages Google Firestore for live data synchronization, GPS-based geofencing for location accuracy, and a dynamic pricing algorithm to optimize slot allocation based on demand and timing. The system enhances user experience by reducing search time and offers parking operators real-time analytics to boost operational efficiency and revenue. Its mobile-first, cloud-native architecture eliminates the need for costly infrastructure, making it a practical and sustainable solution for modern urban environments.

1.1 Background and Basics:

As urban populations grow and private vehicle ownership rises, parking has become a pressing issue for both city planners and daily commuters. Traditional parking methods—such as manual on-street systems or outdated hardware-based setups—struggle to adapt to the fast-paced demands of modern urban environments. These legacy systems often lack flexibility, transparency, and the ability to respond in real time. However, advancements in mobile technology, GPS, and cloud computing have paved the way for smarter parking solutions. Modern real-time parking apps now provide users with accurate, up-to-date information on space availability, while operators gain valuable insights through analytics and revenue management tools. Building on these innovations, the proposed system delivers real-time functionality without relying on costly IoT devices, instead leveraging Firebase Firestore and a modular microservices architecture for scalability and efficiency.

1. 2 Problem Statement:

As vehicle ownership continues to surge, particularly in densely populated and developing urban areas, the challenge of locating available parking spaces has become a major contributor to traffic congestion, fuel wastage, and commuter

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

frustration. Traditional parking systems, reliant on manual monitoring and fixed pricing, lack the agility to meet the demands of dynamic urban environments. Although smart parking solutions using IoT sensors and camera systems offer improvements, their high deployment and maintenance costs limit their scalability and suitability for mid-sized cities. This underscores the need for a low-cost, scalable, and user-friendly parking solution that adapts to real-time demand, enhances operational efficiency, and provides a seamless experience for both users and operators.

1.3 Project Scope:
Optimized Smart Parking System project aims to develop a flexible, real-time, mobile-first platform that improves urban parking experiences. Its scope includes creating an Android application that enables users to discover available parking slots, make reservations, and complete digital payments in real time. A dynamic pricing engine is integrated to adjust rates based on demand and time-related factors, enhancing space utilization. The system uses Google Firebase Firestore to ensure seamless real-time data synchronization between users and service providers. Future iterations will offer parking operators dashboards and analytics tools to manage slot availability and pricing effectively. Additionally, the backend is built on a scalable microservices architecture to support easy integration and maintenance. With a focus on user-centric design and cloud-based technologies, the project aims to reduce search times, optimize parking space usage, and deliver a sustainable, efficient, and cost-effective solution for urban settings.

II. PROJECT PLANNING AND MANAGEMENT

Effective planning and management are vital to the successful development of the Optimized Smart Parking System. This section highlights key strategies, including requirement specification, development methodology, scheduling, and codebase estimation. A structured planning phase ensures alignment between technical goals, real-world deployment, and long-term scalability.

2.1 System Requirements Specification (SRS):
The Smart Parking System is built on a modular client–server architecture optimized for real-time performance and scalability. It connects a native Android app with cloud-hosted backend services and Firebase Firestore for live data synchronization. The mobile app provides users with an intuitive interface to view parking availability, book slots, make payments, and receive notifications. Backend microservices handle core business logic through RESTful APIs, while Firestore ensures low-latency data storage for user profiles, bookings, and slot status. Key features include real-time slot discovery, dynamic pricing based on demand, seamless booking with Razorpay integration, and push notifications for updates. A web-based operator dashboard for slot management and analytics is planned for future development.

2.2 Functional requirements:
The Optimized Smart Parking System is designed to deliver key functional capabilities, including real-time synchronization of parking slot availability using Firestore's listener services. Pricing is dynamically calculated through API interactions that analyze current usage patterns, ensuring demand-based rate adjustments. The system supports secure and atomic workflows for both reservation and payment processes, minimizing errors and ensuring transaction integrity. Additionally, Firebase Cloud Messaging (FCM) is employed to send timely notifications for important user events such as booking confirmations, expirations, and pricing changes.

2.3 Non-functional requirements:
The Optimized Smart Parking System is engineered to meet high non-functional standards. Performance is optimized to ensure that 95% of API responses have a latency of under 300 milliseconds. It supports scalability through auto-scaling mechanisms to accommodate over 1,000 concurrent users without degradation in performance. Security is enforced using OAuth2-based authentication and granular Firestore security rules to protect user data and system integrity. Usability is prioritized with a streamlined three-step booking process, allowing users to complete reservations quickly and intuitively from app launch to confirmation.

2.4 Deployment Environment:
On the client side, the system supports Android devices running version 8.0 or higher with GPS capabilities, utilizing the Firebase SDK for authentication, real-time database access, and cloud messaging. Development and testing tools include Git for version control, Maven for build automation, Postman for API testing, and draw.io for system diagramming. The only external hardware requirement is a GPS-enabled Android smartphone with at least 2 GB of

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

RAM to ensure smooth user experience.

2.5 Software Development Methodology

The project followed an Agile Scrum methodology to support continuous delivery, rapid iteration, and adaptive planning. Development was organized into two-week sprints, with daily stand-up meetings held to monitor progress and address any blockers. A prioritized product backlog, structured around user stories, guided feature development and task allocation. The CI/CD pipeline enabled automated testing and deployment through staging environments, ensuring code quality and stability. Additionally, canary releases were used for gradual rollouts, allowing safe and controlled updates to the production environment.

III. ANALYSIS AND DESIGN

3.1 Introduction:

A robust analysis and well-structured design are essential for the development of a scalable and efficient smart parking system. The Optimized Smart Parking System follows a structured planning methodology that includes strategic prioritization of features, a comprehensive feasibility assessment, and a modular architectural approach. This section outlines the system's conceptual modeling and technical viability.

3.2 Idea Matrix:

To drive innovation and ensure the project meets real-world challenges, a strategic matrix-based analysis was performed. The IDEA, D, E, and A matrices helped refine the project's objectives, guiding system design with measurable impacts.

I – Increase/Input/Impact/Innovation/Ignore

Increase: Accuracy of real-time slot visibility and precision in dynamic pricing.
Input: Driver's GPS data, parking lot metadata, and historical demand patterns.
Impact: Aims to reduce average search time by over 50% and increase parking operator revenue by around 15%.
Innovation: Introduced geofencing-based updates and demand-responsive pricing.
Ignore: Outdated sensor data and support for non-Android platforms.

D – Deliver/Detect/Decrease

Deliver: Smooth booking and payment features, real-time notifications for critical events.
Detect: Entry/exit events using geolocation, demand surges for dynamic pricing.
Decrease: User effort in booking (≤ 3 taps) and search time by half.

E – Eliminate/Educate/Evaluate

Eliminate: Manual counting of slots and use of outdated or stale data.
Educate: Includes onboarding guides for users and operator documentation.
Evaluate: Ongoing performance monitoring through API latency tracking and pricing model testing.

A – Advance/Avoid/Advantage

Advance: Future support for predictive models (e.g., LSTM, ARIMA).
Avoid: Sensor over-dependence and rigid pricing mechanisms.
Advantage: Focus on low-cost deployment, cloud-native infrastructure, and scalable services.

3.3 System Architecture Overview:

The system follows a client-server architecture with a modular, microservices-based backend designed for scalability and real-time data processing. The Android mobile client serves as the user interface, offering live parking slot discovery, reservation functionality, and integrated digital payments. Backend services expose RESTful APIs to handle core functions such as dynamic pricing, booking transactions, session management, and user notifications. Firebase Firestore acts as the cloud-hosted database, providing low-latency data storage and real-time synchronization between users and parking providers. This architecture supports independent development and scaling of each module, simplifying maintenance and enabling future enhancements like AI-powered demand forecasting and multi-tenant provider dashboards.

© 2025 IJMRSET | Volume 8, Issue 6, June 2025| DOI:10.15680/IJMRSET.2025.0806161

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. FLOW OF SYSTEM

The Optimized Smart Parking System operates on a client-server architecture, facilitating seamless communication between the Android mobile app, backend microservices, and Firebase Firestore for real-time data synchronization.

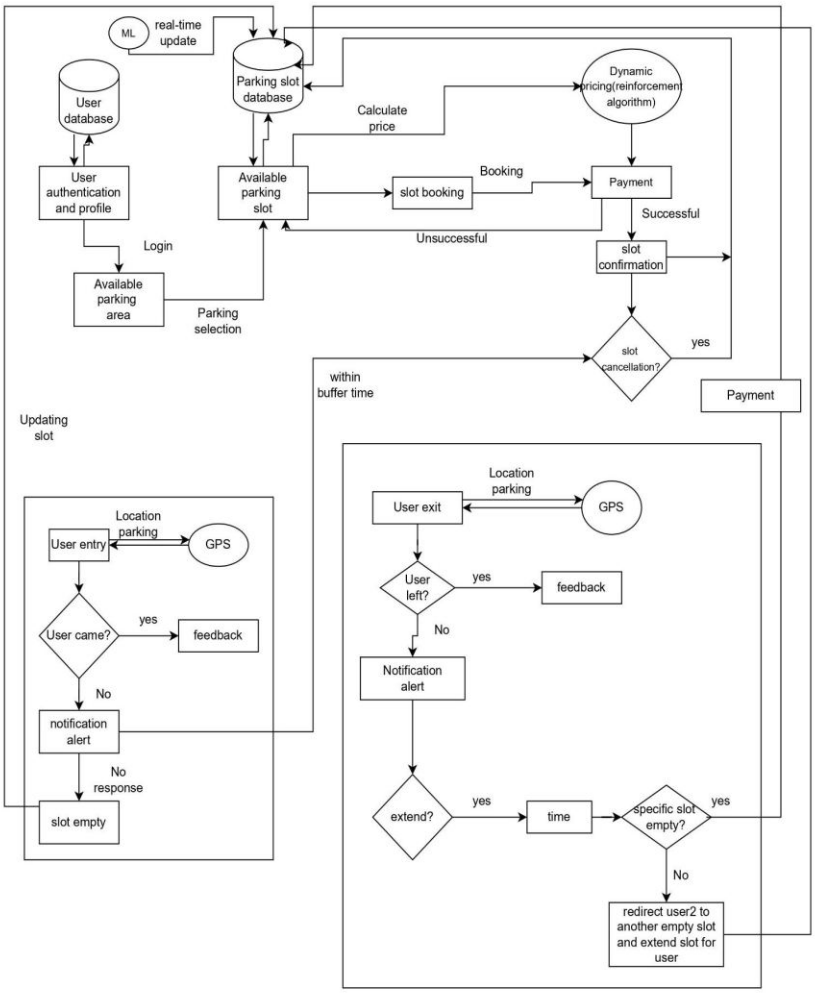
4.1 User Interaction and Slot Discovery:

Users launch the app, which detects their GPS location and retrieves nearby parking availability from Firestore. Slots are displayed on an interactive map with real-time status updates via Firestore's addSnapshotListener(). In offline scenarios, cached data is shown with prompts to reconnect for live accuracy.

4.2 Dynamic Pricing Computation:

Upon selecting a parking slot and duration, the backend calculates pricing dynamically using the formula: price =basePrice×(1+α×(currentDemandhistoricalAverage−1))×Δtprice=basePrice×(1+α×(historicalAverage/currentDemand −1))×Δt

where basePrice is ₹30/hour, α is the demand coefficient (e.g., 1.2), and Δt is the duration in hours. This approach applies surge pricing during peak demand and discounts off-peak, balancing user satisfaction and revenue.



System Flow Diagram

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4.3 Booking and Payment Workflow:

Once a user confirms a slot and time, the backend executes an atomic reservation transaction to prevent double bookings. Payment is processed via Razorpay supporting UPI, cards, and wallets. Successful payments update Firestore slot status to "booked," log a unique transaction ID, and store booking details. Failed payments trigger an automatic rollback and user notification for retry.

4.4 Notification Management:

Firebase Cloud Messaging (FCM) delivers notifications for booking confirmations, slot expiry alerts (5–10 minutes before end), surge pricing alerts, and promotional offers. Each device is registered with a unique token for personalized communication.

4.5 Data Handling and Synchronization:

All data—booking records, user profiles, slot statuses, and pricing—is stored in Firestore's hierarchical NoSQL database, organized as ParkingLots → Slots → {status, slotID, timestamp}. Updates to slot status (e.g., vehicle entry/exit) trigger real-time synchronization across all connected clients, ensuring data consistency and instant UI updates.

V. CONCLUSION

The flow of the system ensures a seamless, real-time experience for both users and parking operators. From live slot discovery and adaptive pricing to secure bookings and instant notifications, each component is tightly integrated to deliver efficiency and ease of use. Real-time data synchronization via Firestore and automated payment handling enhance user convenience, while the modular backend supports scalability and future upgrades. This cohesive design addresses modern urban parking challenges with a smart, cost-effective solution.

REFERENCES

[1] T. Lin, H. Rivano, and F. Le Mouël, "A Survey of Smart Parking Solutions," IEEE Trans. Intell. T ransp. Syst., vol. 18, no. 12, pp. 3229–3253, Dec. 2017.
[2] H. Jung, J. M. Lee, and K. Hwang, "Smart Parking Management System Using AI," Webology, vol.no. 19, no. 1, Art. 4629, Feb. 2022.
[3] R. A. Devi et al., "Smart Car Parking System Using Image Processing and Deep Learning," Int. Res.J. Eng. Technol., vol. 8, Special Issue, Apr. 2021.
[4] J. Parmar, P. Das, and S. M. Dave, "Study on Demand and Characteristics of Parking System in Urban Areas: a Review," J. Traffic Transp. Eng. (Engl. Ed.), vol. 7, no. 1, pp. 111–124, Jan. 2020.
[5] G. Raghuraman et al., "Smart Parking System with Dynamic Pricing Model," Indian J. Sci. Technol., vol. 16, no. 18, pp. 1332–1339, Sep. 2023.
[6] M. G. Díaz Ogás, R. Fabregat, and S. Aciar, "Survey of Smart Parking Systems," Tech. Rep., Nat. Univ. San Juan, 2020.
[7] Y. Xu and Ö. Onan, "IoT-Based Smart Parking Management System," IEEE Internet Things J., vol. 5, no. 3, pp. 1038–1048, Jun. 2018.
[8] B. Zhang, L. Wang, and J. Gao, "A Blockchain-Enabled Parking System with Dynamic Pricing," in Proc. IEEE Blockchain Conf., 2019, pp. 45–52.
[9] T. K. Nguyen, Q. Duong, and H. Le, "Hybrid Sensor Fusion for Smart Parking," Sensors, vol. 21, no. 5, Art. 1593, Mar. 2021.
[10] L. Guo, X. Xu, and K. Zhao, "Predictive Analytics for Parking Availability Using ARIMA and LSTM," J. Urban Data Sci., vol. 3, no. 2, pp. 87–102, Jul. 2021.
[11] M. A. Alsarier and M. M. Saleh, "Smart Parking System Based on IoT Sensor Networks," IEEE Access, vol. 9, pp. 112345–112356, 2021.
[12] B. S. Chowdhury, A. Mukherjee, and R. Basu, "A Review of Dynamic Pricing Strategies in Smart Mobility," Transp. Res. Part C, vol. 132, Art. 103430, Jun. 2022.