



RESEARCH ARTICLE

Parking System: A Comprehensive Management Solution for Small Parking Facilities

Claudisnei ^{1*}

¹ SourceCodester Community, Brazil.

Correspondence

^{1*} SourceCodester Community, Brazil.
Email: claudisnei@sourcecodester.com

Funding information

SourceCodester Community, Brazil.

Abstract

This study presents the design and implementation of a Parking Management System specifically developed for small-scale parking facilities. The system aims to automate core operations such as vehicle tracking, rate calculation, and payment processing to reduce inefficiencies common in manual management. Developed using Delphi and the Paradox database, the system applies a modular two-tier architecture integrating independent modules for vehicle entry, exit, billing, and reporting. Data validation, automated fee computation, and structured reporting features ensure accurate transactions and transparency. The interface is designed for non-technical users, emphasizing clarity, reliability, and operational speed. Implementation results show improved efficiency, reduced processing time, and enhanced financial accuracy. Despite technological constraints related to scalability and legacy database architecture, the system remains a cost-effective and practical solution for small parking operators. Future improvements could include migration to modern relational databases, integration with mobile and web platforms, and adoption of emerging technologies such as IoT sensors and AI-based analytics to support smart, data-driven parking management.

Keywords

Parking Management; Automation; DELPHI; Database System; Small Facilities.

1 | INTRODUCTION

Parking management systems play a vital role in maintaining the functionality of urban infrastructure, particularly in small facilities that often operate with constrained budgets and limited technical capacity. These operations typically rely on manual procedures that are prone to inefficiency and revenue leakage, which can diminish customer satisfaction and profitability. Recent studies emphasize that automation and digital tools can substantially reduce such inefficiencies. For instance, the integration of QR code-based systems has been shown to enhance accuracy in data recording and streamline payment processes, reducing dependence on manual intervention (Ingole *et al.*, 2023). Likewise, the development of intelligent parking systems incorporating wireless sensor networks and artificial intelligence enables real-time monitoring of parking availability, guiding drivers to open spaces more efficiently while mitigating congestion (Srivastava *et al.*, 2023; Tang *et al.*, 2006; Sri *et al.*, 2024). These innovations are particularly valuable for smaller facilities,

as they deliver operational improvements without the need for extensive infrastructure (SaraMucheka & Mupeta, 2025). From an economic standpoint, digitalization in parking management contributes directly to increased revenue and operational transparency. A large-scale analysis conducted in Tanzania found that the adoption of adaptable smart parking frameworks significantly improved income management for operators, highlighting the financial viability of technological modernization even in resource-limited contexts (Kusyama *et al.*, 2024). Parallel research underscores the environmental dimension of smart parking systems, noting their potential to reduce carbon emissions through improved traffic flow and efficient space utilization, aligning parking management with sustainable urban development goals (Sakib *et al.*, 2024; Takaendengan *et al.*, 2024). Operational strategies within parking facilities are also influenced by their underlying business models. Traditional hourly and daily pricing mechanisms are often preferred in compact facilities, balancing user affordability and space turnover (Caicedo *et al.*, 2006; Sándor & Csiszár, 2015).

Fixed daily rates simplify transactions, while subscription-based models create predictable revenue streams and customer loyalty (SaraMucheka & Mupeta, 2025). Premium valet systems further diversify offerings, demanding precise coordination of service quality and vehicle handling (Faheem *et al.*, 2013; Lam & Yang, 2019). In recent years, hybrid models integrating multiple pricing strategies have become increasingly common, supported by cloud-based management and mobile applications that dynamically adjust rates according to demand and occupancy levels (Ingole *et al.*, 2023; Wang *et al.*, 2023). Such flexibility allows operators to respond to fluctuating market conditions while maintaining service reliability. To meet these operational, financial, and environmental demands, the proposed Parking System is designed to automate key management functions without imposing excessive technical complexity. It records vehicle entries and exits with time-stamped precision, automatically calculates fees according to variable rate structures, and accommodates supplementary services such as vehicle washing. The system also incorporates receipt generation, payment tracking, and reporting functions to assist in daily accounting and decision-making. Developed using Delphi and a Paradox database, the system offers a practical, low-cost solution for small-scale operators seeking dependable management tools without the overhead of enterprise software. By combining automation, flexibility, and simplicity, it provides a realistic technological framework for small parking facilities striving for efficiency, transparency, and sustainability.

2 | SYSTEM ARCHITECTURE AND DESIGN

The Parking System was developed with a focus on operational reliability, scalability for small facilities, and ease of maintenance. It integrates a structured technological foundation, a well-defined database schema, a modular application framework, and an interface optimized for usability. Together, these components support seamless management of vehicle entries and exits, payment processing, service tracking, and reporting, ensuring that parking operations remain efficient and consistent even with limited technical resources. The system is implemented using Delphi, a rapid application development (RAD) environment widely recognized for building database-oriented business applications. Delphi compiles native code that delivers fast performance and uses a mature Visual Component Library (VCL) to create responsive and visually coherent Windows-based interfaces. Its strong database connectivity through the Borland Database Engine (BDE) and integrated report generation tools make it ideal for small-scale operational systems. Complementing Delphi, the system employs the Paradox database as its storage backend. Paradox is a file-based database system that operates without a dedicated server, offering simple deployment, easy maintenance, and sufficient performance for small to medium datasets. Although Paradox is considered legacy technology, its compatibility with Delphi, absence of runtime licensing costs, and minimal hardware requirements make it an effective solution for single-user or small multi-user parking environments. Structurally, the system adopts a traditional two-tier client-server model, where the application layer manages presentation and business logic while the database layer handles data storage and retrieval. This design is practical for local operations, offering stability without the complexity of web-based infrastructure. The database design forms the backbone of the Parking System, ensuring that all operational data is systematically stored, processed, and retrieved. Key entities include the *Vehicles* table, which records essential information such as license plate numbers, vehicle types, colors, and optional owner details; the *ParkingSessions* table, which documents each parking transaction from entry to exit; and the *Payments* table, which logs all financial transactions, including parking fees, additional services, total amounts, and payment methods. Additional tables such as *Services*, *RateStructure*, and *Employees* support service management, pricing flexibility, and user administration. Supplementary structures including *Configuration*, *AuditLog*, and *Reports* are used for system settings, activity tracking, and data reporting. Referential integrity is enforced through primary and foreign key relationships, while constraints, triggers, and indexing ensure data consistency, efficient querying, and historical traceability for auditing and analysis. The application is built with a modular architecture to promote maintainability and scalability.

Each functional module operates independently while remaining integrated within the overall system workflow. The main dashboard serves as the central hub, displaying key operational indicators such as parking occupancy, revenue summaries, and recent transactions. The *Vehicle Entry* module manages the registration of incoming vehicles, generates

unique ticket numbers, and records entry data automatically. The *Vehicle Exit* module facilitates check-out procedures by retrieving vehicle details, calculating parking fees based on predefined rate structures, processing payments, and issuing receipts. Supporting these are the *Payment Processing* module, which handles diverse payment methods and transaction tracking, and the *Service Management* module, which records optional services like vehicle washing or detailing. Additional modules such as *Search and Inquiry* and *Reporting* enable quick information retrieval and provide analytical insights into performance metrics, revenue streams, and employee productivity. Administrative features encompass rate configuration, user management, database maintenance, and audit review. This modular design simplifies troubleshooting, allows isolated testing, and facilitates future system expansion. The user interface is designed with emphasis on clarity, accessibility, and operational speed to support non-technical users. The main dashboard provides real-time summaries of occupancy, revenue, and transactions, with prominently displayed action buttons for essential tasks such as new entries, exits, searches, and report generation. Data entry screens employ structured input fields with validation mechanisms for license plate accuracy, vehicle type selection, and optional details like color or model. Checkout screens automatically calculate duration-based fees and display itemized charges to ensure transparency. The reporting interface enables the generation of daily summaries, occupancy analyses, and financial overviews, which can be printed or exported to common formats such as PDF and Excel. Administrative pages allow easy modification of rates, user credentials, and system configurations through guided menus and confirmation prompts. Consistency in layout, font clarity, and color-coded functions enhance operator efficiency, while keyboard shortcuts and validation rules minimize input errors. Overall, the system architecture combines proven technologies and structured design principles to deliver a functional and maintainable parking management solution. By balancing simplicity with operational depth, it supports small parking facilities in achieving automation, accuracy, and reliability without requiring extensive technical infrastructure.

3 | CORE FUNCTIONALITIES

The Parking System is designed to streamline the full cycle of parking operations, from vehicle entry registration to financial reporting, through a combination of automation, structured data handling, and user-oriented functionality. Each process is integrated into a cohesive workflow that minimizes manual effort, reduces operational errors, and enhances service consistency for both operators and customers. The system begins operation when a vehicle arrives at the parking facility. The operator initiates the entry procedure by launching the entry module and recording the vehicle's license plate number, which the system automatically formats and validates. It then checks the database to determine whether the vehicle is a returning customer, retrieving stored information when available or creating a new record for first-time entries. The operator can select the vehicle type from predefined options, add optional details such as color, model, or owner's name, and, if applicable, assign a parking spot. Once the data is confirmed, the system generates a unique ticket number—either sequentially or as a barcode—records the exact time of entry, and creates a new parking session in the database with an active status. A printed parking ticket containing the entry details and pricing information serves as proof for both the operator and the customer. This entire process is optimized for speed, typically requiring less than thirty seconds to complete, ensuring efficiency during peak periods. When a customer exits, the system retrieves the corresponding active record by searching the database using the license plate or ticket number. The interface displays the complete session data, including vehicle details, entry and exit times, duration of stay, and assigned parking spot. The system then calculates the total fee automatically, applying either hourly or daily rates as defined by the active pricing configuration. It accounts for variables such as time of day, day of the week, vehicle type, and applicable discounts. For customers who have used additional services, such as car washing or detailing, the related charges are added to the final bill. The operator inputs the payment amount and selects the payment method, after which the system calculates any change due and finalizes the transaction. A detailed receipt is printed, including an itemized list of fees, payment information, and duration of stay. Once the transaction is completed, the parking session status is updated to "Completed," the space is marked as available, and the database reflects the new occupancy count. The process is designed to be completed within one to two minutes, ensuring both accuracy and customer satisfaction.

Pricing within the system is managed through a flexible rate configuration framework that supports hourly, daily, weekly, and monthly billing schemes. Administrators can define multiple rate structures, including variations by day, time, or vehicle type, and apply rules such as grace periods, rounding methods, or daily caps. The rate calculation engine dynamically selects the appropriate pricing structure based on the session's attributes, ensuring accurate and consistent billing. The system also maintains a complete history of rate changes for auditing and analytical purposes, with built-in validation to prevent conflicting configurations and to warn administrators if changes affect active sessions. In addition to parking fees, the system manages optional services that provide added value and revenue streams. Operators can record customer requests for services such as washing, interior cleaning, or minor maintenance directly within the parking session. The system stores service details, timestamps, and associated costs, which are automatically integrated into the customer's final bill. Service completion status can be tracked by staff to ensure accountability and quality control. These records also contribute to reporting and analysis, allowing management to assess service utilization, identify popular

offerings, and adjust pricing or promotions accordingly. By incorporating service management, the system not only supports operational efficiency but also broadens the facility's business potential. Efficient data retrieval is a central feature of the Parking System. The search module allows operators to quickly locate records through multiple criteria, including license plate number, ticket number, date range, status, or vehicle type. Partial matches, wildcards, and case-insensitive searches are supported to improve flexibility. Search results are displayed in an interactive grid that can be sorted, filtered, or exported for reporting and documentation purposes. The advanced search option allows users to combine multiple conditions using logical operators, while built-in indexing and optimized queries ensure fast response times even as the database grows.

The reporting and analytics component converts operational data into actionable insights that support decision-making and performance evaluation. Standard reports include daily revenue summaries, occupancy trends, employee productivity, service revenue breakdowns, and analyses by vehicle type. Reports can be customized to specific time periods, filtered by criteria such as payment method or session type, and exported to formats like PDF or Excel. Visual elements such as charts and graphs enhance interpretability, and automated scheduling enables reports to be generated and distributed at regular intervals. Through these analytical capabilities, managers can assess performance, identify peak usage patterns, optimize pricing, and forecast capacity requirements. The administrative functions of the system provide full control over user management, system configuration, and data security. Administrators can create and modify user accounts, assign roles and permissions, and enforce password policies to maintain system integrity. The configuration panel allows adjustments to operational parameters such as facility information, parking layout, ticket templates, and rate structures. Regular database maintenance features include backup, restore, and optimization tools to ensure data reliability. An integrated audit log records all user activities, including logins, modifications, and transactions, allowing for accountability and compliance monitoring. The system also implements data encryption, automatic session timeouts, and health monitoring tools to track system performance and resource utilization. Overall, the core functionalities of the Parking System demonstrate a well-integrated approach to operational automation. By combining structured data management, efficient transaction processing, and analytical reporting, the system delivers a complete framework that enhances transparency, reduces administrative workload, and supports informed decision-making. Its modular yet interconnected design ensures that all aspects of parking management—entry, exit, billing, services, and administration—operate seamlessly within a single cohesive platform.

4 | RESULTS AND DISCUSSION

4.1 Discussion

The Parking System was successfully implemented as a Windows-based desktop application using Delphi's Visual Component Library (VCL). The development process adhered to Delphi's best practices by maintaining a clear separation between user interface, data logic, and database interaction. Core components such as TForm for windows, TDataSource for data binding, and TDBGrid for tabular display were used extensively to deliver a responsive, visually consistent user experience. Event-driven programming ensured smooth user interaction through actions like OnClick, OnChange, and keyboard shortcuts, while robust exception handling preserved stability during runtime. The modular code architecture consists of a main control form serving as the system hub and several submodules for entry, exit, reports, search, and administrative functions. The backend was implemented using the Paradox database, organized as a set of .DB files in a dedicated directory. Each table was designed and indexed to support fast retrieval and integrity constraints, including primary and foreign key enforcement. The Borland Database Engine (BDE) served as the connectivity layer between Delphi and Paradox, with TTable components used for simple CRUD operations and TQuery components for executing parameterized SQL statements. Transaction handling ensured data consistency using the standard StartTransaction, Commit, and Rollback structure, while exception management caught and handled database-specific errors gracefully. Figure 1 illustrates the system architecture that governs data control and transaction flow between modules.

CONTROLE DE ESTACIONAMENTO

Serviço Consulta

Novo Grava Cancela Entrada Fecha Anula Cadastro Back-up Localizar Placa

Entrada N° Placa Veículo Espécie
 Hora Diária

ESTACIONAMENTO

Tipo
 Particular
 Convênio
 Pré-Pago

Limpeza
 Interna Completa
 Externa Lavação...

Complemento
 Cêra
 Motor
 Serv. Terceiros

Data Entrada Cliente Diárias Cêra
Hora Entrada Qtd. Dias Horas Motor
Data Saída Qtd. Hora Itens Serv. Terceiros
Hora Saída Sub-Total Descontos Total Geral

Figure 1. System architecture for parking space management

Printing functionality was integrated to automate the issuance of parking tickets and receipts, an essential operational requirement. Tickets were formatted to include the facility header, ticket number, vehicle details, entry timestamp, and optional barcode for future integration with scanning systems. Receipts contained comprehensive details—entry and exit times, calculated duration, itemized charges, and payment information—to ensure financial transparency. The system used fixed-width fonts, bold emphasis, and spacing alignment for readability, while print preview and reprint options enhanced usability. To preserve data accuracy, the system applied multiple layers of validation and error handling. Input validation verified license plate formatting, data completeness, and numeric field correctness. Business rule validation prevented duplicate tickets, double assignments of parking spots, or incomplete payments. Database constraints and triggers maintained referential integrity, while descriptive error messages guided operators in resolving issues. All critical code blocks were enclosed in exception-handling structures (try-except), with logs automatically generated for troubleshooting. These validation and recovery mechanisms ensured operational continuity even under system stress or user error. The deployment of the system demonstrated tangible benefits for parking facility operators, including reduced manual workload, faster entry and exit cycles, and improved accuracy in billing. Automation eliminated arithmetic errors, maintained precise time tracking, and standardized rate applications. Revenue monitoring became more transparent through consistent fee calculations, recorded audit trails, and daily reporting. Operational efficiency improved with shorter customer wait times and higher throughput, while customers benefited from professional printed documentation, transparent pricing, and faster service.

From a managerial perspective, the system facilitated detailed business intelligence by capturing data across operational, financial, and personnel dimensions. Reports generated insights on occupancy trends, revenue breakdowns, employee activity, and service utilization. This data enabled better decision-making for pricing strategies, capacity management, and staff allocation. Additionally, the system's cost-effectiveness proved advantageous: it requires no licensing fees, minimal infrastructure, and low maintenance overhead while yielding rapid return on investment through improved operational accuracy and customer retention. The deployment and maintenance phase followed a structured implementation process. Installation required configuring the BDE alias, setting up printer drivers, and creating an administrator account. Daily operational routines included data entry, session management, and end-of-day reconciliations supported by automated revenue reports. Weekly and monthly maintenance involved reviewing statistics, optimizing the database, and backing up data. Backup and recovery mechanisms employed full and incremental copies of database files, ensuring data restoration in case of

corruption or hardware failure. Troubleshooting procedures covered common issues such as printer malfunctions, database locks, and configuration errors, each addressed through detailed user documentation and recovery guides.

4.2 Result

The implementation results demonstrate that the Parking System delivers significant operational improvements; however, several limitations constrain its scalability and modernization potential. The Paradox database, while efficient for small-scale deployments, lacks support for concurrent access beyond two or three users and is limited in handling very large datasets. Its legacy architecture and dependency on the Borland Database Engine restrict compatibility with modern systems. Moreover, as a desktop-based application, the system requires installation on each workstation and lacks remote access or mobile compatibility. This architecture, while adequate for small facilities, presents barriers to distributed or enterprise-level deployment. Operationally, the system is tailored for single-location environments without centralized management across multiple facilities. It lacks advanced features such as online reservations, mobile payments, or integration with automated gate systems. Security is limited to basic username–password authentication, with no granular role-based permissions or encrypted backups. Furthermore, reporting is restricted to predefined templates without interactive visualization or real-time monitoring dashboards. While sufficient for routine operations, these limitations reduce flexibility and responsiveness in data-driven management scenarios. Scalability challenges are inherent to the underlying file-based architecture. As historical data accumulates, query performance and data retrieval speed diminish. The system’s design does not support horizontal scaling, clustering, or high-availability configurations. Expansion to multiple locations requires independent installations, leading to fragmented datasets and manual consolidation. These challenges indicate that modernization would be necessary for broader operational deployment or integration with external systems.

Looking ahead, modernization could be achieved through technical upgrades such as migrating the database to modern relational systems like MySQL, PostgreSQL, or SQL Server. This transition would enhance concurrency, support large datasets, and improve transaction reliability. Developing a web-based interface using frameworks such as ASP.NET, Laravel, or Node.js would allow browser-based and mobile access, while cloud hosting on platforms like AWS or Azure would ensure scalability and remote management. Additionally, introducing RESTful APIs would enable integration with external applications such as accounting, property management, or customer relationship management (CRM) systems. In terms of feature evolution, the system could incorporate online reservations, digital payments, automated gates, and loyalty or discount programs to enhance customer convenience and engagement. Integrating with mobile applications would extend accessibility, allowing users to reserve spots, view charges, or make payments remotely. Enhanced analytics dashboards and real-time data visualization would improve managerial oversight and decision-making. Multi-location support could centralize data collection and reporting, allowing unified control over several facilities. Emerging technologies offer transformative potential for the system’s future. License Plate Recognition (LPR) using computer vision could automate entry and exit validation, reducing operator intervention. IoT sensors could provide real-time occupancy data, predictive maintenance alerts, and heat maps for space utilization. Artificial intelligence could optimize dynamic pricing, forecast demand, and detect anomalies in transactions. Furthermore, blockchain integration could secure payments through immutable records, while electric vehicle (EV) support would align the system with the growing shift toward sustainable mobility. These innovations, when combined, could evolve the current solution into a next-generation smart parking platform capable of autonomous operations and data-driven optimization.

5 | CONCLUSIONS AND RECOMMENDATION

The Parking System developed in this study provides a practical and effective approach to managing small-scale parking facilities through automation of vehicle tracking, billing, and payment processes. Designed using Delphi and the Paradox database, the system demonstrates that dependable and affordable technological solutions can still be achieved with legacy platforms when applied to well-defined operational contexts. It addresses the most common challenges faced by small parking operators—manual inefficiencies, calculation errors, and revenue leakage—by offering automated functionality within a user-friendly interface that requires minimal technical expertise. The combination of intuitive design, accurate fee computation, structured reporting, and additional service management makes the system a comprehensive tool for daily operations. The system’s key advantages lie in its reliability, operational simplicity, and low implementation cost. It successfully streamlines the full parking workflow—from vehicle entry and exit to payment processing and reporting—while maintaining high accuracy and efficiency. Operators benefit from clear and professional ticketing, consistent application of rate structures, and the ability to generate analytical reports for business insights. These strengths not only improve operational

performance but also enhance customer satisfaction through faster service and transparent billing. Although the platform lacks modern features such as real-time web access, mobile integration, or cloud scalability, these limitations are acceptable trade-offs for its intended users: small, independent facilities that prioritize affordability and ease of use over advanced technological sophistication. In essence, the Parking System represents a sound example of problem-oriented software engineering. It underscores that effective digital transformation in small businesses does not necessarily depend on cutting-edge technology but on the alignment between system capability and user need. For operators of modest parking facilities, this system provides a viable entry point into digital management practices, building both technical familiarity and data awareness. Moreover, the operational experience and historical data collected through the system can later support migration to more advanced, networked, or cloud-based platforms as facilities expand. From an implementation standpoint, prospective users should assess their specific operational scale, resource availability, and infrastructure readiness before adopting the system.

The software is best suited for facilities with under one hundred parking spaces, a limited number of concurrent users, and a Windows-based computing environment. Successful deployment depends on management commitment, adequate hardware preparation, staff training, and gradual adoption of advanced features. Operators are advised to begin with core functionalities—entry, exit, and billing—before integrating additional services and reporting tools. Continuous monitoring of performance and user satisfaction will help identify opportunities for refinement and ensure sustainable system use. For developers or organizations customizing the software, a thorough understanding of its architecture, database schema, and business logic is essential. Enhancements can focus on modernizing the data layer by migrating from Paradox to SQL-based databases, improving the user interface, or integrating new modules such as web access or mobile applications. Adopting contemporary software engineering practices—version control, modular design, automated testing, and comprehensive documentation—will also improve maintainability and scalability. Developers are encouraged to share improvements and documentation to foster collaborative growth and community-based innovation. At the industry level, the project highlights several broader insights for parking management. It illustrates how appropriately scaled technology can enhance efficiency without imposing the financial and technical burdens of enterprise-level systems. The future of parking management will likely revolve around trends such as contactless transactions, mobile and online reservations, integration with urban mobility platforms, and sustainability-driven operations. Industry stakeholders should therefore approach technological investment with a balanced view—valuing practicality and long-term viability as much as innovation. Collaborative initiatives to establish data standards, interoperability frameworks, and shared development efforts can further support the evolution of accessible, intelligent parking solutions. Overall, the Parking System stands as a demonstration of how targeted, well-implemented technology can deliver measurable improvements in operational control, financial accuracy, and customer experience for small parking facilities. It bridges the gap between manual management and advanced automation, paving the way for broader digital adoption within the parking sector while maintaining a focus on simplicity, reliability, and real-world usability.

ACKNOWLEDGMENTS

Appreciation to Claudisnei for developing and sharing this practical parking management solution, the SourceCodester community for providing a platform for knowledge sharing, Embarcadero Technologies (formerly Borland) for Delphi development tools, the parking industry professionals who inspire practical solutions, and all small business operators who benefit from accessible technology.

REFERENCES

- Alarbi, M., Jaradat, A., Lutfiyya, H., & Haque, A. (2023). Scope: Smart cooperative parking environment. *IEEE Access*, *11*, 120346–120361. <https://doi.org/10.1109/access.2023.3328233>
- Access*, *11*, 120346–120361. <https://doi.org/10.1109/access.2023.3328233>
- Caicedo, F., Robusté, F., & López-Pita, A. (2006). Parking management and modeling of car park patron behavior in underground facilities. *Transportation Research Record: Journal of the Transportation Research Board*, *1956*(1), 60–67. <https://doi.org/10.1177/0361198106195600108>

- Cantu, M. (2012). *Database application development with Delphi*. Embarcadero Press.
- Embarcadero Technologies. (n.d.). *Embarcadero Delphi documentation*. <https://docwiki.embarcadero.com/>
- Faheem, M., Mahmud, S., Khan, G., Rahman, M., & Zafar, H. (2013). A survey of intelligent car parking systems. *Journal of Applied Research and Technology*, 11(5), 714–726. [https://doi.org/10.1016/S1665-6423\(13\)71580-3](https://doi.org/10.1016/S1665-6423(13)71580-3)
- Ingole, M., Patle, C., Gahane, S., Atkare, A., & Bante, A. (2023). QR-based car parking system. *International Research Journal of Modernization in Engineering, Technology and Science*. <https://doi.org/10.56726/irjmets47293>
- International Parking Institute. (2015). *Parking management best practices*.
- Kusyama, S., Magemo, A., & Giga, M. (2024). Critical analysis of parking systems and their economic impact in Tanzania. *MJRD*, 5(4), 1–10. <https://doi.org/10.62277/mjrd2024v5i40070>
- Lam, P., & Yang, W. (2019). Application of technology to car parking facilities in Asian smart cities. *Journal of Facilities Management*, 17(2), 142–156. <https://doi.org/10.1108/JFM-05-2018-0031>
- Mahmud, S., Khan, G., Rahman, M., & Zafar, H. (2013). A survey of intelligent car parking system. *Journal of Applied Research and Technology*, 11(5), 714–726. [https://doi.org/10.1016/S1665-6423\(13\)71580-3](https://doi.org/10.1016/S1665-6423(13)71580-3)
- Paradox Database Documentation. (n.d.). *Corel Corporation* [Archived].
- Sakib, N., Bakibillah, A., Susilawati, S., Kamal, M., & Yamada, K. (2024). Eco-friendly smart car parking management system with enhanced sustainability. *Sustainability*, 16(10), 4145. <https://doi.org/10.3390/su16104145>
- Sándor, Z., & Csiszár, C. (2015). Role of integrated parking information system in traffic management. *Periodica Polytechnica Civil Engineering*, 59(3), 327–336. <https://doi.org/10.3311/ppci.7361>
- SaraMucheke, S., & Mupeta, M. (2025). Design and development of a web-based car park management system. *International Journal of Advances in Engineering and Management (IJAEM)*, 7(3), 654–663. <https://doi.org/10.35629/5252-0703654663>
- Sridhar, K., Neha, K., Sudheshna, K., Gayathri, M., & Bhanurangarao, M. (2024). Robust parking space allocation system using OpenCV and scikit-learn. *Journal of Image Processing and Intelligent Remote Sensing*, 43, 37–46. <https://doi.org/10.55529/jipirs.43.37.46>
- Srivastava, A., Dwivedi, A., Maheshwari, A., Rajput, A., & Gupta, D. (2023). Enhancing urban living with automatic parking management systems: A comprehensive review and case study. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 191–199. <https://doi.org/10.32628/cseit2390411>
- Takaendengan, T., Rangkang, J., Sudarno, S., & Soukotta, D. (2024). Planning campus parking facilities and infrastructure: Towards sustainable mobility approaches. *European Modern Studies Journal*, 8(1), 50–58. [https://doi.org/10.59573/emsj.8\(1\).2024.6](https://doi.org/10.59573/emsj.8(1).2024.6)
- Tang, V., Zheng, Y., & Cao, J. (2006). An intelligent car park management system based on wireless sensor networks. *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 65–70. <https://doi.org/10.1109/SPCA.2006.297498>
- Thompson, R. (2011). *Desktop database applications: Design and implementation*. Tech Publishing.
- Wang, Y., Wang, T., & Li, D. (2023). Design and implementation of an Android-based intelligent parking reservation management system. *Journal of Information and Knowledge Engineering*, 1(4), 57–66. <https://doi.org/10.62517/jike.202304412>

How to cite this article: Claudisnei, C. (2025). Parking System: A Comprehensive Management Solution for Small Parking Facilities. *Journal Dekstop Application (JDA)*, 4(1). <https://doi.org/10.59431/jda.v4i1.661>.