УДК 004.67:932.519.6
DOI 10.58649/1694-9099-2026-1-484-488

РАШИТОВА А.Ж., ЧЖАН ТАО
Жусуп Баласагын атындагы КУУ,
И. Арабаев атындагы Кыргыз мамлекеттик университети
РАШИТОВА А.Ж.,[1] ЧЖАН ТАО[2]
КНУ имени Жусупа Баласагына,[1]
Кыргызский государственный университет имени И. Арабаева[2]
RASHITOVA A.ZH., ZHANG TAO
KNU Jusup Balasagyn,
Kyrgyz State University named after I. Arabaev

0009-0003-2643-1546[1]

PYTHON ТИЛИНДЕ QR-КОДДОРДУ ГЕНЕРАЦИЯЛОО СИСТЕМАСЫН ДОЛБООРЛОО

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ГЕНЕРАЦИИ QR-КОДОВ НА PYTHON

DESIGN OF A QR CODE GENERATION SYSTEM IN PYTHON

Кыскача мүнөздөмө: Макалада Python тилинде QR-коддорду генерациялоо системасын долбоорлоо маселеси функционалдык жана функционалдык эмес талаптарды системалуу талдоонун негизинде каралат. Талаптардын ортосундагы карама-каршылыктарга, атап айтканда, ката оңдоо деңгээли, маалымат сыйымдуулугу, коопсуздук жана өндүрүмдүүлүк ортосундагы тең салмакка өзгөчө көңүл бурулат. Талаптарды приоритеттештирүүнүн зарылдыгы көп критерийлүү чечим кабыл алуу ыкмалары, анын ичинде fuzzy TOPSIS аркылуу негизделет. Системанын негизги функционалдык мүмкүнчүлүктөрү, анын ичинде ар түрдүү маалымат форматтарын колдоо, генерация параметрлерин жөндөө жана пакеттик иштетүү, ошондой эле туруктуулук, өндүрүмдүүлүк жана коопсуздук сыяктуу функционалдык эмес мүнөздөмөлөр каралат. Изилдөөнүн актуалдуулугу санариптик трансформация шартында жана QR-технологияларды билим берүү жана административдик процесстерде колдонуу контекстинде негизделет. Натыйжалар системаны долбоорлоодо структураланган мамиленин маанилүүлүгүн жана анын натыйжалуулукка жана ишенимдүүлүккө тийгизген таасирин көрсөтөт.

Аннотация: В статье рассматривается проектирование системы генерации QR-кодов на языке Python на основе системного анализа функциональных и нефункциональных требований. Особое внимание уделяется выявлению противоречий между требованиями, таким как баланс между уровнем коррекции ошибок, ёмкостью данных, безопасностью и производительностью системы. Обоснована необходимость приоритизации требований с использованием методов многокритериального анализа, в частности fuzzy TOPSIS. Описаны ключевые функциональные возможности системы, включая поддержку различных типов данных, параметров генерации и пакетной обработки, а также нефункциональные характеристики, такие как устойчивость, производительность и безопасность. Показана актуальность разработки в контексте цифровой трансформации и применения QR-технологий в образовательных и административных процессах. Полученные результаты демонстрируют значимость структурированного подхода к проектированию и его влияние на повышение эффективности и надёжности системы.

Abstract: The article examines the design of a QR code generation system in Python based on a structured analysis of functional and non-functional requirements. Particular attention is paid to identifying conflicts between requirements, such as the trade-off between error correction level, data capacity, security, and system performance. The necessity of requirement prioritization using multi-criteria decision-making methods, particularly fuzzy TOPSIS, is substantiated. The paper describes key functional capabilities of the system, including support for multiple data types, configurable generation parameters, and batch processing, as well as non-functional characteristics such as robustness, performance, and security. The relevance of the study is demonstrated in the context of digital transformation and the application of QR technologies in educational and administrative processes. The results highlight the importance of a structured design approach and its impact on system efficiency and reliability.

Негизги сөздөр: QR-код; Python; программалык инженерия; функционалдык талаптар; функционалдык эмес талаптар; fuzzy TOPSIS; санариптик трансформация.

Ключевые слова: QR-код; Python; программная инженерия; функциональные требования; нефункциональные требования; fuzzy TOPSIS; цифровая трансформация.

Keywords: QR code; Python; software engineering; functional requirements; non-functional requirements; fuzzy TOPSIS; digital transformation.

The design of a QR code generation system in Python begins with a systematic and structured analysis of requirements, as the system being developed must satisfy both functional tasks (defining its behavior) and non-functional constraints (characterizing the quality of its performance). In software engineering theory, requirements define the boundary conditions of the system architecture and directly influence the design decisions made.

In practice, functional and non-functional requirements often come into conflict. For example, increasing the level of error correction improves the robustness of a QR code against damage, but simultaneously reduces the available data capacity. Similarly, the implementation of additional security mechanisms enhances data protection but increases computational load and may affect system response time. Therefore, requirements analysis should not be limited to simple enumeration; it must include prioritization, trade-off evaluation, and architectural justification of design decisions [10].

From a methodological perspective, a structured specification of requirements reduces uncertainty at the implementation stage and ensures logical consistency between theoretical principles and practical solutions applied in system development.

Relevance in the context of digital infrastructure

The significance of developing a QR code generation system extends beyond individual applied solutions and is determined by broader processes of digital transformation. In the Kyrgyz Republic, particularly within the framework of the interdepartmental electronic interaction system “Tunduk,” digital mechanisms for identification and data verification are being actively implemented. In this context, QR codes are becoming an important tool for document verification, ensuring digital authentication of services, as well as organizing educational and certification processes [4].

Although the proposed system does not предполагает direct integration with the national digital infrastructure, its architectural solutions comply with key requirements of digital governance, including interoperability, reliability, and data security. This allows the developed system to be considered as a universal model potentially applicable in broader digital ecosystems.

Functional requirements

At the functional level, the proposed QR code generation system based on Python must implement the complete encoding pipeline defined by the ISO/IEC 18004 standard, including data mode selection, bit stream construction, data padding, error correction code generation, matrix placement, masking, and final image rendering. The primary functional requirement is the accurate and standardized generation of QR codes based on user input. However, for practical scalability and real-world applicability, the system must go beyond basic encoding capabilities. It should support multiple payload types (numeric, alphanumeric, byte/UTF-8), allow the selection of QR code versions and error correction levels (L, M, Q, H), and provide configurable output parameters such as image format (PNG/SVG), size, resolution, and quiet zone dimensions.

In addition, to ensure robustness and prevent invalid outputs, the system must include input validation mechanisms, such as capacity overflow detection and encoding compatibility checks [9]. In educational and administrative contexts—such as classroom testing, student registration, or digital attendance tracking—additional functional requirements include batch generation of QR codes, automatic assignment of identifiers, and optional integration with structured data sources (e.g., CSV or Excel files) [1]. In such scenarios, the system functions not merely as a QR image generator but as a component of a broader identification and workflow management process, thereby expanding its functional role within digital infrastructures.

Furthermore, the system should be designed with extensibility in mind, allowing future integration with external services, APIs, or institutional platforms. This includes the potential for incorporating authentication mechanisms, database connectivity, and support for dynamic QR codes that can redirect to updated content. Such enhancements would further strengthen the applicability of the system in real-world digital ecosystems and increase its practical value.

Non-functional requirements define the quality attributes under which the QR code generation system must operate and are particularly critical, as QR technologies are often deployed in time-sensitive and reliability-critical environments. In educational and administrative contexts, QR codes must be quickly and accurately scanned under various real-world conditions, including uneven lighting, varying camera resolutions, screen glare, or partial physical occlusion.

Thus, robustness becomes a key non-functional requirement, directly linked to the proper selection of error correction levels and masking strategies defined in ISO/IEC 18004, as well as adherence to readability principles such as sufficient quiet zones and high contrast ratios. Performance efficiency is equally important: the system must support fast generation, especially in batch processing scenarios, while maintaining consistent results across different operating systems and Python environments [5].

Security and privacy considerations also represent significant non-functional constraints, particularly when QR codes encode personal identifiers or institutional data. The exposure of raw payload information may lead to data leakage risks; therefore, requirements analysis must anticipate architectural solutions such as identifier hashing, minimization of logging, and avoidance of embedding sensitive data directly within QR codes. Additionally, the system should consider secure data transmission practices and potential encryption mechanisms where applicable.

Importantly, non-functional requirements frequently conflict with functional goals. For instance, higher levels of error correction (Q or H) improve robustness but reduce data capacity; stronger encryption enhances security but may complicate decoding processes; and increased configurability improves flexibility but may negatively affect system performance. These inherent trade-offs necessitate a structured prioritization mechanism to balance competing quality attributes rather than treating all requirements as equally important [2].

Such prioritization ensures architectural consistency and supports well-justified design decisions during subsequent stages of system implementation [6]. Moreover, it contributes to the development of a balanced and efficient system capable of operating effectively in diverse real-world conditions while meeting both user expectations and technical constraints.

Use of fuzzy TOPSIS for requirement prioritization (methodological justification)

For the systematic prioritization of functional requirements while balancing non-functional constraints, studies in requirements engineering often employ multi-criteria decision-making (MCDM) methods. Among them, fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is particularly suitable in contexts where evaluation criteria are not strictly quantitative and may involve ambiguity or expert judgment, such as “high usability” or “medium security risk” [7]. The fundamental principle of TOPSIS consists in ranking alternatives by measuring their relative closeness to the ideal solution (representing optimal performance across all criteria) and their distance from the negative ideal solution (representing the worst-case scenario). The fuzzy extension enhances this framework by allowing the use of linguistic variables and imprecise evaluations, which are common in requirements assessment processes [8].

In the context of a QR code generation system, fuzzy TOPSIS can be applied as a conceptual prioritization framework to determine which requirements should guide architectural and design decisions. For example, candidate requirements may be evaluated according to criteria such as core functional importance, impact on user experience, implementation complexity, security significance, and performance implications. By structuring requirement comparisons through a multi-criteria framework, potential conflicts—such as robustness versus capacity or security versus usability—can be analyzed more transparently.

Even if the present study does not implement a full computational model of fuzzy TOPSIS, referencing this well-established methodology enhances the academic rigor of the requirements analysis. It demonstrates that prioritization decisions are not arbitrary but are grounded in recognized principles of decision-making theory within software engineering research [8].

Practical outcomes of prioritization for the system

Based on the domain-specific characteristics of QR code technologies and their typical implementation scenarios, several requirements emerge as priorities within the hierarchy. First and foremost, correctness and strict compliance with the ISO/IEC 18004 standard constitute the primary requirement, as deviations from the standard directly threaten compatibility and decoding reliability [3]. Closely related to this is robustness under real-world scanning conditions, including the proper implementation of quiet zones, appropriate masking strategies, and the selection of suitable error correction levels to ensure readability despite environmental variability [3].

Configurability of error correction levels and output parameters is also critically important, as operational contexts—such as printed materials, mobile screens, or projection displays—impose different constraints. In addition, batch processing capabilities become a priority requirement in educational and administrative environments, where QR codes are often generated for multiple users simultaneously within structured workflows.

Finally, a privacy-aware design approach is required to mitigate risks associated with embedding sensitive data, favoring the use of internal identifiers or hashed references instead of directly exposing personal information within the payload. In contrast, advanced user interface enhancements, support for additional barcode standards, or complex external integrations represent lower-priority features, as they do not affect the correctness of the system core and can be implemented incrementally.

This prioritization structure reflects the principles of incremental system development, in which standard compliance and reliability are established first, followed by usability enhancements and automation improvements.

Thus, the analysis of requirements for a QR code generation system should not be limited to a simple listing of characteristics. The design process must account for the interdependencies and conflicts between functional and non-functional requirements, while ensuring a well-justified prioritization structure. The application of structured methodologies, particularly fuzzy TOPSIS, provides a theoretically grounded tool for ranking competing requirements and minimizing conflicts between system objectives [8].

The use of this approach enhances methodological transparency and strengthens the scientific validity of design decisions. Moreover, it enables the establishment of a conceptual link between theoretical foundations—including QR code structure, error correction mechanisms, and decoding robustness—and the practical architectural solutions implemented in the system [3; 4].

The design of a QR code generation system in Python requires a comprehensive and structured approach to requirements analysis, integrating both functional capabilities and non-functional quality attributes. The study has demonstrated that the effectiveness of such a system depends not only on accurate implementation of encoding processes in accordance with the ISO/IEC 18004 standard but also on the ability to balance competing requirements, including robustness, data capacity, security, and performance.

The analysis highlights the importance of identifying and managing inherent trade-offs between system characteristics. In this context, the application of multi-criteria decision-making approaches, particularly fuzzy TOPSIS, provides a theoretically grounded framework for prioritizing requirements and supporting informed design decisions. Even at a conceptual level, this methodology contributes to increased methodological rigor and transparency in the system design process.

Furthermore, the results emphasize that a well-prioritized requirement structure enables the development of a scalable, reliable, and adaptable system suitable for real-world applications, including educational and administrative environments. The proposed approach ensures consistency between theoretical principles and practical implementation, thereby enhancing the overall quality and applicability of the system.

In conclusion, the integration of structured requirements analysis with systematic prioritization methods represents a key factor in the successful design of QR code generation systems, contributing to their efficiency, reliability, and relevance within modern digital infrastructures.

References

1. Tribak, H., Zaz, Y. QR Code recognition based on principal components analysis method // International Journal of Advanced Computer Science and Applications (IJACSA). – 2017. – Vol. 8, № 4. – P. 242–248. – DOI: 10.14569/IJACSA.2017.080433.
2. Kusyanti, A., Arifin, A. QRishing: a user perspective // International Journal of Advanced Computer Science and Applications (IJACSA). – 2017. – Vol. 8, № 10. – P. 302–309. – DOI: 10.14569/IJACSA.2017.081039.
3. ISO/IEC 18004:2015. Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification. – Geneva : International Organization for Standardization, 2015.
4. Reed, I. S., Solomon, G. Polynomial codes over certain finite fields // Journal of the Society for Industrial and Applied Mathematics. – 1960. – Vol. 8, № 2. – P. 300–304. – DOI: 10.1137/0108018.
5. Ohbuchi, E., Hanaizumi, H., Hock, L. A. Barcode readers using the camera device in mobile phones // Proceedings of the 2004 International Conference on Cyberworlds. – Washington : IEEE, 2004. – P. 260–265. – DOI: 10.1109/CW.2004.67.
6. Lin, Y., Fuh, C. S. 2D barcode image decoding based on adaptive thresholding and perspective correction // IEEE Transactions on Image Processing. – 2013. – Vol. 22, № 11. – P. 4351–4364. – DOI: 10.1109/TIP.2013.2278010.
7. ZXing Project. Zebra Crossing: multi-format 1D/2D barcode image processing library [Электронный ресурс]. – 2024. – Режим доступа: https://github.com/zxing/zxing
8. Li, J., Bian, W., Diao, Y., Zou, T., Yang, X., Kang, B. A multi-user quiz system with QR code identification and configurable timing // Applied System Innovation. – 2025. – Vol. 8, № 6. – Art. 158. – DOI: 10.3390/asi8060158.
9. Pressman, R. S., Maxim, B. R. Software engineering: a practitioner's approach. – 9th ed. – New York : McGraw-Hill Education, 2019.
10. Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. – 3rd ed. – Upper Saddle River : Prentice Hall, 2004.

Рецензент: к. социолог. н., доцент Шайылдаева А.К.