PRISHTINA DIGITAL WALLET

Prishtina Digital Wallet Software Analysis

1 Introduction

The Prishtina Digital Wallet project aims to integrate various municipal services into a single application, enhancing convenience and efficiency for the citizens of Prishtina. This analysis outlines the software requirements, architecture, and key components necessary for the successful implementation of the Prishtina Digital Wallet.

By digitizing invoicing and payment procedures, the Prishtina Digital Wallet greatly decreases paper consumption, helping to reduce waste and the necessity for physical travel, minimizing the carbon footprint associated with municipal access and improving environmental sustainability.

2 Project Scope

The application will enable citizens to access municipal services, manage and pay invoices from different municipal sources, and interact with public and future private sector services online. Furthermore, the digital wallet will not be the only function of the system. The system will evolve to integrate other municipal services, such as education services, applying for registration in particular school as well as for first level of health services.

3 System Overview

3.1 User Identification and Authentication

Enhanced security by reducing the needs for different passwords, utilizing difficult to replicate factor and providing a layered defense against unauthorized access.

- To streamline user access by connecting the digital wallet to the Prishtina Single Sign-On (SSO) service, enabling one set of login credentials for multiple services.
- The system will integrate with the Prishtina SSO service, allowing users to log in using their Prishtina municipal account credentials. This integration facilitates a unified access point for various municipal services, enhancing user experience.
- Incorporate biometric authentication like fingerprint scanning, facial recognition, or iris scanning, tailored to the device’s capabilities, to provide a quick and secure access mechanism.
- Alongside Prishtina SSO and biometric verification, MFA will be used, incorporating a mix of knowledge-based (passwords or PINs), possession-based (a trusted device), and inherence-based (biometrics) methods.

User accounts must be opened through a trusted service operator that has rights to offer these services in Kosovo and in European Union.

The system shall ensure the privacy and provide means of user consent and control over the use of biometric and personal data, storage and protection, aligning with privacy regulations, such as GDPR.

The system shall employ robust encryption for secure storage and transmission of authentication data to prevent data breaches.

3.2 Invoice Aggregation

The Invoice Aggregation system within the Prishtina Digital Wallet will centralize the collection and management of invoices from various sources, facilitating a streamlined and user-friendly experience for the citizens.

This component shall be able to expand easily in the future and add new utilities, by simply registering and providing API keys for further access to upload or download the data from the system.

- Ensure the aggregation system is scalable to accommodate the addition of new services and invoice sources in the future. Design the system to be flexible, allowing for the easy integration of new municipal or public enterprise services as they become digitalized.

Invoice data correlated with the system can be internal or external.

- In the case of internal systems, an interface shall be developed to connect with internal municipal billing systems, ensuring that invoices generated by different departments are automatically imported into the Digital Wallet Platform.
- In the case of the external system the system shall establish secure data exchange protocols to retrieve and integrate their billing information into the system. The established data exchange protocols can be a formatted and structured data in xml, json, csv and other acceptable formats.

In each case the data should have mandatory fields and optional fields.

The fields and their determination shall be done by developing team during the analysis phase.

Each actor that interacts with the system, shall have a full notification of the invoices imported and their status of importing process.

- Provide municipal administrators with tools to manage the integration settings, monitor data flows, and troubleshoot issues between the Digital Wallet and internal or external invoicing systems.

The system shall enable bulk and transactional importing of invoices.

3.3 Invoice and User Correlation

User account correlation with available invoices and the future imported one shall be enabled through two methods:

- Personal Number: Where available, use citizens' personal identification numbers to automatically match and correlate invoices to their respective Digital Wallet accounts, ensuring that users can view and manage all their municipal invoices in one place.
- Client ID: For services where personal numbers are not used, implement a system where users can link their accounts with client IDs provided by the service entities. Allow users to manually correlate invoices to their Digital Wallet account by entering the last invoice number or reference, facilitating the integration of services into the unified billing platform. Security measures shall be employed to verify the identity of users attempting to link invoices or client IDs to their Digital Wallet account, to prevent unauthorized access or any fraudulent activity.

A notification system shall be developed to alert users of new invoices, successful linkages, or any issues in correlating invoices with their Digital Wallet account, enhancing transparency and trust in the system.

3.4 Invoice Management

The Invoice Management component of the Prishtina Digital Wallet will serve as a central hub for citizens to manage and pay their municipal invoices, encompassing services like utilities, taxes, urban traffic ticket, parking fees and other fines.

The invoice management component shall be designed in an easy-to-navigate interface that allows users to quickly view, manage and pay invoices with minimal effort. Is shall have a dashboard that highlights key information and as well the upcoming events and notification system.

The Invoice Management component shall be accessible through web portal and mobile application.

Features and Functionalities

- Invoice Aggregation: The system will consolidate invoices from various municipal departments and public enterprises, providing a unified view of all outstanding and paid invoices for each citizen.
- Real-time Invoice Updates: Implement real-time data syncing to ensure that invoice statuses are updated promptly, reflecting payments, adjustments, or new charges, allowing citizens to see their current financial obligations at any time.
- Invoice Details and History: Users will have access to detailed information for each invoice, including itemized charges, due dates, payment history, and any applicable discounts or penalties.
- Payment Processing: Integrate with secure payment gateways to enable direct payment of invoices through the app. This feature should support various payment methods, including credit/debit cards, bank transfers, and mobile payment systems.
- Automated Reminders and Notifications: Set up automated reminders for upcoming due dates and notifications for newly issued invoices or receipt of payment, ensuring that users stay informed and avoid late payments.

- Dispute Resolution and Support: Provide a mechanism for users to dispute inaccuracies or issues with invoices directly through the app, including a ticketing system for tracking the resolution process.

3.5 Wallet Balance

The Wallet Balance feature within the Prishtina Digital Wallet shall allow users to maintain a virtual balance that can be used to pay for municipal services directly within the app, offering a convenient and streamlined financial management tool. Key functionalities and features of the Wallet Balance shall be:

- Top-Up Facility: Users can add funds to their digital wallet via bank transfers, credit/debit cards, or at designated physical locations, providing flexibility in how they manage their wallet balance.
- Balance Management: The wallet provides real-time information on the current balance, recent transactions, and spending history, helping users track their expenditures and manage funds effectively.
- Auto-Pay Options: Enables users to set up automatic payments for recurring expenses like utilities or property taxes, ensuring timely payments and avoiding late fees.
- Budgeting and Alerts: Offers budgeting tools within the wallet, allowing users to set spending limits for different services and receive alerts when they approach these limits.

Users shall be able to send and receive funds to other users through their wallet balance.

3.6 Payment System

The Payment System in the Prishtina Digital Wallet facilitates transactions for various municipal services, enabling users to pay directly through the app using their preferred payment method. Core functionalities required are:

- Supports of payment options, including direct payments from the digital wallet balance, and direct payment with credit/debit cards. The payment system shall be designed to be intuitive and user-friendly, allowing for quick and hassle-free payments with just a few clicks.
- Transactions shall be processed in real-time, ensuring immediate reflection in service status and account balances, providing users with up-to-date financial information. Users shall easily access their payment history and download receipts for each transaction, aiding in financial tracking and record-keeping.
- Collaborates with reputable payment gateways to ensure that all transactions are securely processed, employing industry-standard security measures like SSL encryption and tokenization.
- Mobile Applications should support the payment with VISA and Mastercard debit and credit cards. The usage of payment through the cards should support the one-time verification and processing through secure tokenized access, will complete payment with one touch.

3.7 Municipal Services Access

Municipal Services Access in the Prishtina Digital Wallet aims to provide a unified portal where citizens can access and pay for various municipal and potentially private sector services, enhancing the convenience and efficiency of urban living.

Current Services

- Urban Traffic Tickets: The wallet will facilitate the payment of traffic tickets, with features to view detailed ticket information, validity period and legal notes. Users can purchase their tickets directly through the app, including categorized tickets as for students, pensioners, daily tickets, weekly, monthly or yearly tickets. Users shall be able to purchase more than one ticket, usually when in bulk, they can only be standard tickets, not for different categories.
- Parking Services: Integrates with the city's parking management system to allow users to pay for parking fees, renew parking permits, and even find available parking spots in real-time, reducing the hassle of physical payments and improving the parking experience.
- Cultural Site Entrances: Offers a platform for citizens to purchase tickets for museums, galleries, and other cultural sites. The wallet can provide digital tickets, which can be scanned for entry, facilitating paperless transactions and promoting cultural engagement.
- Municipality fines: Users shall be able to pay also fines, as parking fines, and other fines that will be available in the system. The system shall process the fines in real time and provide means of notification to the issuing entity.

System shall be developed in such a way that future expansion will be available through configuration, without a need for further development.

Integration with Private Sector Services:

- Taxi Services: Plans to include payment options for taxi fares within the app, allowing users pay for rides seamlessly, enhancing the public transportation experience.
- Bars and Restaurants: Future updates may allow users to pay for meals and drinks at participating bars and restaurants, offering a cashless and convenient dining experience.

3.8 Notification system

The system shall implement a comprehensive notification framework that includes push notifications, in-app notifications, and email notifications. Each service and feature will be seamlessly integrated into the notification system, providing users with the ability to customize their notification preferences. Users will have options to mute notifications, apply temporary muting periods, and access other configuration settings to tailor their notification experience according to their needs.

Key Features:

Push Notifications: Real-time alerts delivered directly to the user's device.

In-App Notifications: Notifications displayed within the application to inform users of updates or actions required.

Email Notifications: Scheduled or triggered emails to keep users informed about important activities and updates.

Customization Options:

Muting: Ability to permanently disable specific notifications.
Temporary Muting: Set time periods during which notifications are paused.
Configuration Settings: Additional options to adjust notification frequency, types, and delivery methods.

3.9 Reporting

Reporting is a crucial component of the solution, as it shall provide insights, data analytics, tracking and auditing solutions.
Each component shall have its adequate initial reports and a tool for further reports design and their publishment shall be available with the system.
Reports shall be possible to be designed for dynamic period selection, entity, user, type or category of transaction.

4 System Architecture

The solution architecture for the Digital Walled is designed to ensure high availability, scalability, and reliability. The architecture will incorporate load balancing, replication, and automated backup systems to minimize downtime and maintain optimal system performance. The architecture is required to be based on web solution, with web services, API and other web components. There are no requirements for programming language.

Key Components

4.1 Load Balancing:

Should be able to integrate and implement a load balancing system to distribute incoming traffic and requests evenly across multiple application servers.

Utilize a combination of hardware and software-based load balancing solutions to ensure optimal performance, fault tolerance, and scalability.

Support both horizontal and vertical scaling to accommodate fluctuations in traffic and system load.

4.2 Replication:

Implement a replicated system architecture, where multiple instances of the application and database servers are deployed across different physical or virtual machines.

Utilize synchronous or asynchronous replication methods, depending on the organization's needs and acceptable latency levels.

Ensure that the replicated instances are kept up-to-date with the latest application and database changes, minimizing the risk of data loss or inconsistencies.

4.3 Automated Backup System:

Develop and deploy an automated backup system to create regular backups of the application and database data.

Store backups in secure, off-site locations to protect against data loss due to hardware failure, natural disasters, or other unexpected events.

Implement backup retention policies and schedules to ensure that the organization's data retention and compliance requirements are met.

Test and verify the backup and restore process periodically to ensure data integrity and successful recovery in case of a disaster.

4.4 High Availability:

Design the architecture with redundancy and failover mechanisms to minimize single points of failure and ensure high availability.

Implement health monitoring and automatic failover processes for application and database servers, so that in case of a failure, the system can automatically switch to a healthy instance.

Utilize clustering and/or container orchestration solutions (e.g., Kubernetes) to manage application instances and maintain high availability.

4.5 Security

Data Security: Implement industry-standard encryption methods to protect data at rest and in transit. Deploy firewalls, intrusion detection, and prevention systems to safeguard the application and database servers against unauthorized access and potential threats.

Network Security: Design a secure network architecture with segmentation, isolating critical components and sensitive data from public networks. Implement virtual private networks (VPNs) or other secure communication channels for remote access and administration.

4.6 Mobile Applications

The Mobile Applications component of the Prishtina Digital Wallet should be developed using native, Flutter, or React Native frameworks to support both iOS and Android platforms. These applications are designed to provide users with a seamless and intuitive experience when accessing municipal services and managing payments.

Development Approach

- Native Applications: Develop native apps for iOS and Android to leverage the full capabilities of each platform, ensuring optimal performance and a superior user experience. This approach allows access to device-specific features such as GPS, camera, and biometric sensors.

- React Native Option: Consider using React Native to build a cross-platform application. This choice aims to streamline development across iOS and Android platforms while maintaining a high level of user experience consistency and reducing time-to-market.
- Flutter Option: Also considered for building a cross-platform application. This framework supports the goal of streamlining development across iOS and Android while maintaining a high level of user experience consistency and reducing time-to-market.

Key Features

- User-Friendly Interface: Design the applications with a focus on ease of use, ensuring that users of all technical skill levels can navigate and manage services effectively.
- Real-Time Synchronization: Implement real-time data synchronization to ensure that the wallet information is always current, reflecting the latest transactions, service updates, and notifications.
- Offline Access: Provide limited offline functionality, allowing users to view their invoice history, ticket information, and other essential data even without an internet connection.

Security and Performance

- Advanced Security Measures: Incorporate industry-standard security practices, including data encryption, secure login mechanisms, and biometric authentication, to protect user information and transactions.
- Optimized Performance: The applications shall be optimized for speed and performance, minimizing load times and enhancing the overall user experience.

4.7 Card support

The system should allow physical card usage by including some functionalities that treat each card as an application registered user. The cards could be topped up at various payment locations and then used for municipal services such as urban traffic, utility payment at payment stations, and parking at various payment stations. The card itself should be based on MIFARE secure technology, or at least MIFARE PLUS technology.

4.8 Mobile Hardware

Mobile hardware is a mandatory for the whole approach solution as it will provide few services for remote areas and service access for the citizens without digital wallet. Such services are, the payment with debit/credit cards on spot, check-up of QR codes of purchased tickets within the digital wallet. Mobile devices should support Barcode and QR code reading, should support and act as a bank POS system for featuring payments with credit/debit cards and should be able to connect to the server for processing payments, checking the validity of tickets and synchronizing data.

5 Features and Functionalities

5.1 Registration and Account Management

- Streamlined registration process with options to link existing municipal service accounts.
- Account management features to update personal information, manage authentication methods, and review transaction history.

5.2 Security and Compliance

- Compliance with data protection regulations (e.g., GDPR).
- End-to-end encryption for data transmission and storage.

5.3 Integration with Municipal Systems

- Seamless integration with existing municipal systems for real-time data exchange and service access.

5.4 Customer Support and Services

- In-app support features, including live chat, FAQs, and a ticketing system for customer inquiries and issue resolution.

6 Technical Requirements

6.1 Data Privacy and Compliance

Data privacy is a crucial aspect of the system, and the registration module as well as invoice aggregation should be designed with data privacy in mind. The following are some potential features that should be included to ensure data privacy:

1. Data Collection: The registration module should only collect the necessary user data required for registration and verification purposes. The system should not collect any unnecessary or sensitive data.
2. Data Storage: The system should store user data securely, protecting it from unauthorized access, modification, or disclosure. The system should use encryption and security protocols to transmit and store user data.
3. Data Use: The system should only use user data for the intended purposes of the case tracking management system. User data should not be shared to third parties without the user's explicit consent.
4. Data Retention: The system should only retain user data for as long as necessary to achieve the intended purposes of the case tracking management system. After that, the system should securely and permanently delete the user data.

5. Data Access: The system should limit access to user data to authorized personnel who require it to perform their duties. The system should use access control measures to ensure that only authorized personnel can access user data.
6. Compliance: The system should comply with relevant data protection laws and regulations, such as GDPR. The system should also have a data protection officer or equivalent to ensure compliance and handle data protection-related queries or complaints.

GDPR: The General Data Protection Regulation (GDPR) is a European Union regulation that sets out strict requirements for the processing of personal data. The Internal User Management module should be designed to comply with the GDPR by:

1. Obtaining consent from users before processing their personal data.
2. Providing users with access to their personal data and the ability to correct or delete it.
3. Limiting the processing of personal data to what is necessary for the specific purpose for which it was collected.
4. Taking steps to protect personal data from unauthorized access, use, or disclosure.
5. Scalable infrastructure to support user growth and service expansion.

6.2 Multilingual Support

The Multilingual Support module is required to enable the Digital Walled to support multiple languages, initially three, with the flexibility to easily add more languages in the future. The module will provide a dynamic system for translating visual components, managing language templates, and allowing users to switch between languages seamlessly.

1. Language Selection:
    a. Provide an intuitive user interface for users to select their preferred language within the application.
    b. Support automatic language selection based on the user's browser settings, location, or other relevant factors, with the option for manual override.
    c. Store user language preferences for future sessions.
2. Dynamic Translation of Visual Components:
    a. Implement a system for dynamically translating all visual components within the application, including menus, buttons, labels, and tooltips.
    b. Allow for easy updating of translations, either manually or through automated processes, to accommodate changes in the application or improvements in translation quality.
3. Language Templates:
    a. Develop a system for managing language templates, which define the layout, formatting, and text elements for documents and reports generated by the case management solution.
    b. Enable users to select and apply language-specific templates based on their language preferences.

    c. Support the creation and modification of language templates by authorized users or administrators, including the ability to add new templates or update existing ones as needed.

4. Extensibility and Scalability:
    a. Design the Multilingual Support module to easily accommodate additional languages in the future, by simply adding new translations and language templates.
    b. Ensure that the module can scale to handle multiple languages without negatively impacting the application's performance or usability.
5. Language Usage Statistics:
    a. Track and record language usage statistics, such as the number of users per language, frequency of language changes, and most common language preferences.
    b. Generate reports on language usage statistics for analysis and decision-making purposes.

By incorporating these specifications into the development of the Multilingual Support module, the Digital Walled will be equipped to support multiple languages and provide a user-friendly, adaptable system for managing language templates, and user preferences.

6.3 Technical Support and Maintenance

The system provider should offer ongoing technical support, updates, and maintenance to ensure the system remains functional, secure, and up-to-date.

Technical Support and data integrity, security, Reporting, and Maintenance for DBMS: The system provider should offer ongoing technical support, updates, and maintenance to ensure the DBMS remains functional, secure, and up-to-date. The technical support should be available 24/7 to help users and administrators troubleshoot problems. The updates should be released regularly to fix bugs and security vulnerabilities. The maintenance should be performed regularly to keep the system running smoothly.

Monitoring and Maintenance

1. Performance Monitoring: Implement monitoring tools and processes to track system performance, resource utilization, and application response times. Set up alerts and notifications for critical events and potential performance issues, enabling proactive troubleshooting and resolution.
2. Maintenance process: Implement a change management process to ensure smooth deployment of updates and minimize the impact on system availability and performance.

3. Penetration Test: At least three months after the system is put to work, EO shall do a penetration test of the system and provide a detailed report to CA. Every issue found on the report will need to be addressed accordingly during the maintenance period.

Minimum maintenance requirements:

1. Highly critical serveries:
    a. EO shall be able to respond to online requests within 1 working hour, for highly critical severity. Resolution should be not higher than 4 hours.
    b. EO shall be able to respond to onsite requests within 4 working for highly critical severities. Resolution should be not higher than 8 working hours.
2. Critical severities, but non-blocking
    a. EO shall be able to respond to online requests within 2 working hours. Resolution shall not exceed 2 working days.
    b. OE shall be able to respond to onsite requests within 8 working hours. Resolution shall not exceed 5 working days.
3. Non critical severities.
    a. EO shall be able to respond to online requests within 4 working hours. Resolution shall not exceed 5 working days.
    b. EO shall be able to respond to onsite requests within 3 working days. Resolution shall not exceed 15 working days.

6.4 Training and Documentation

The solution should come with comprehensive documentation and training materials to help users understand and effectively use the system. Training materials shall cover all the modules, system overview and each used third-party components.

Trainings shall be held in joint session for

- Solution overview

And in separate sessions for

- Administrators of the system
- Managers and users of departments and external units in a separate session.

6.5 Source Code

The project deliverables must include the source code, intended exclusively for the internal use of the municipality. This provision is set to ensure that the municipality can independently manage system updates and upgrades if required, particularly in circumstances such as the economic operator's bankruptcy.

Source code should be provided with documentation.

However, it is expressly stated that the intellectual property rights associated with the solution remain with the Economic Operator. This includes the exclusive rights to sell and resell the solution in the open market. This arrangement is designed to maintain the balance between the agency's need for operational security and the Economic Operator's intellectual property rights.

6.6 Project Management

For the successful completion of this software project, it is mandated to use Agile project management methodology. Agile promotes a flexible, collaborative, and customer-focused approach to software development, which aligns perfectly with our objectives for this project. The following are the key elements of an agile project management approach.

1. Iterative Development: The project shall be managed into short, manageable segments known as "sprints." Each sprint will have a clear set of deliverables and will typically last 1-2 weeks. At the end of each sprint, there will be a product increment that adds value to the software. It is the bidders responsibility to provide the project schedule.
2. Prioritizations: The project shall be prioritized based on the requirements from the Contracting Authority, where some deliverables may be provided in advance and with higher priority.
3. Collaboration and Communication: Agile methodology encourages continuous collaboration among project stakeholders. This includes daily stand-up meetings for team members to discuss their progress and address any challenges. Stakeholders and end users will also be involved throughout the project, ensuring that their feedback is integrated into the development process.
4. Flexibility and Responsiveness: One of the primary strengths of Agile methodology is its ability to adapt to changes. Bidders must understand that requirements may evolve during the course of the project. Agile allows to modify the direction of the project as necessary to accommodate changing requirements while still maintaining progress towards overall goals.
5. Quality Assurance: Testing is integrated throughout the Agile development process, rather than being relegated to the end of the project. This means that issues can be detected and rectified early on, ensuring the quality of the final product. However, the final testing and acceptance will be mandatory.
6. Transparency: Agile approach promotes transparency at all levels of the project. All team members and stakeholders will have access to up-to-date information about the project's status, progress, obstacles, and next steps.
7. Risk Management: Agile methodology allows risks to be identified and addressed early on in the project lifecycle, helping to avoid major issues down the line. It is the bidders responsibility to implement Risk Management Methodology and provide with Risk Identification and mitigation actions.
8. Continuous Improvement: After each sprint, the developer team shall conduct a "retrospective" meeting to discuss what went well, what didn't, and how we can improve in the next sprint.
9. Delivery and Deployment: With Agile, working software is delivered frequently, providing more opportunities for feedback and ensuring that the product aligns closely with the needs and expectations of end users.

Project should be managed on phases, giving the priority to the modules required by the Institution. Such modules and iterations should be released for public use on the Appstore and Google Play Store.

7 Project Deliverables

The minimum requirements as project milestones are as follows:

#	Project deliverable	Time
1.	Compliance Matrix	Bid
2.	Conceptual Solution and architecture:	Bid
3.	Conceptual Project Management Plan:	Bid
4.	Risk Identification Methodology:	Bid
5.	Conceptual Training Plan:	Bid
6.	Support and Maintenance Plan:	Bid
7.	High Level Design: Implementation Phase -	20 days after signing the contract
8.	Review of the software analysis, detailing and compatibility with the new organizational structure at the time of signing the contract	20 days after signing the contract.
9.	System Architecture: Implementation Phase	20 days after signing the contract
10.	Database Conceptual Design: Implementation Phase	20 days after signing the contract
11.	Detailed Project Management Plan: Implementation Phase	20 days after signing the contract
12.	UI/UX proposal, mock-ups	Each Iteration Phase
13.	Data Migration	During Iteration Phase involved with data migration
14.	Deliverables and testing	Each Iteration Phase
15.	Low Level Design	Each Iteration Phase
16.	Data Privacy Compliance	Each Iteration Phase
17.	Detailed Training Plan and Documentation: Before Final Acceptance	Shall not exceed 5 months from contract signature.
18.	Detailed Testing Plan: Before Final Acceptance of solution	20 days before the planned Final Acceptance
19.	Detailed Support and Maintenance Plan: Before Final Acceptance of solution	20 days before the planned Final Acceptance
20.	Source Code with documentation: After Final Acceptance of solution	Shall not exceed 6 months from contract signature.
21.	Penetration Test: After Final Acceptance of solution	90 days after Final Acceptance

Consideration to release the first module for Traffic Urban ticketing before 1st of January 2025 is a mandatory. The proposed project timelines should include and consider this release.

8 Checklist & Evaluation Criteria

The Prishtina Digital Wallet is envisioned as a comprehensive solution to modernize and streamline the interaction between the citizens of Prishtina and municipal services, laying the foundation for future digital integration with the private sector.

#	Criteria	CRITERIA	Weight	FORMULA
1.	Compliance Matrix	PASS / FAIL	1	
2.	Solution Architecture	PASS / FAIL	1	
3.	Project Management Plan	PASS / FAIL	1	
4.	Risk Identification Methodology	PASS / FAIL	1	
5.	Training Plan	PASS / FAIL	1	
6.	Support and Maintenance Plan	PASS / FAIL	1	
7.	Customer Support Plan	PASS / FAIL	1	
8.	Payment Gateway Processor	PASS / FAIL	1	
9.	Project Manager	PASS / FAIL	1	
10.	Software Analyst	PASS / FAIL	1	
11.	Publication & Advertising Plan	PASS / FAIL	1	
12.	Sales Plan & Stakeholders engagement	PASS / FAIL	1	
13.	Financial analysis and Business Model Assessment	PASS / FAIL	1	
14.	Transaction fee for payments (excluding Prishtina Parking and Urban Traffic Tickets)	Fixed fee	30%	$F = \frac{F_l}{Fa} \times 30\%$
15.	Transaction % for payments of Urban Traffic Tickets	%	15%	$T = \frac{T_l}{Ta} \times 15\%$
16.	Transaction % for payments for Prishtina Parking	%	15%	$P = \frac{P_l}{Pa} \times 15\%$
17.	Transaction % for payments through Deit/Credit Card Processors	%	5%	$D = \frac{D_l}{Da} \times 5\%$
18.	Transaction % for payments processing through third financial institution parties and Top Up Points for City cards	%	5%	$X = \frac{X_l}{Xa} \times 5\%$
19.	Customer Support Lines 24/7	No of Lines	10%	$L = \frac{L_a}{L_h} \times 10\%$
20.	Implementation Period	Months	10%	$I = \frac{I_l}{Ia} \times 10\%$
21.	Contracting Period after implementation	Minimum Contracting Years	10%	$C = \frac{C_a}{C_h} \times 10\%$

F_l = Lowest Transaction Fee
F_a = Tender Transaction Fee
T_l = Lowest Percentage Commission for Urban Traffic Tickets

T_a = Tender Percentage Commission for Urban Traffic Tickets
D_l = Lowest Percentage Commission for Credit/Debit card processors
D_a = Tender Percentage Commission for Credit/Debit card processors
X_l = Lowest Percentage Commission for processing payments through third party financial institutions
X_a = Tender Percentage Commission for processing payments through third party financial institutions
P_l = Lowest Percentage Commission for Prishtina Parking Tickets
P_a = Tender Percentage Commission for Prishtina Parking Tickets
I_l = Lowest Implementation Period
I_a = Tender Implementation Period
L_h = Highest number of 24/7 customer support lines
L_a = Tender number of 24/7 customer support lines
C_h = Highest number of contracting years
C_a = Tender number of contracting years

9 Eligibility Criteria

Experience in e-Wallet development for payment transfer, interconnected with State Treasury (end-to-end solution) – at least one reference.

Agreement with State Treasury for payment processing for Public Institutions with at least 3 years of experience.

Bidders should be certified with ISO standards:

- ISO 9001:2015
- ISO 14764
- ISO 27001

At least 250,000 Euro of revenues for each consecutive year in last three years on software development.

One (1) Project Manager – with at least bachelor degree on Computer Sciences or Computer Engineering. Project Manager should be certified in .NET framework or equivalent certification, Certified Project Management Professional from PMI or equivalent, with at least 7 years of experience. Copies of CV, Diplomas, References and Certificates should be provided.

One (1) Business Analyst – with at least university bachelor degree on Business Administration with at least 3 years of experience on systems similar to e-Wallet. Copies of CV, Diplomas, References and Certificates should be provided.

One (1) System Architect – with at least bachelor degree on Computer Sciences or Computer Engineering, certified on .Net framework or equivalent, with at least 5 yearson development of Large-Scale Applications. Copies of CV, Diplomas, References and Certificates should be provided.

One (1) Quality Control Expert – with at least bachelor degree on Computer Sciences or Computer Engineering, certified on .Net framework or equivalent, with at least 5 yearson development of Large-Scale Applications. Copies of CV, Diplomas, References and Certificates should be provided.

Four (4) Software Developers – with at least bachelor degree on Computer Sciences or Computer Engineering, certified as full-stack software developers. Software developers should be experienced in at least three (3) implementation of e-Wallet system developments. Copies of CV, Diplomas, References and Certificates should be provided.

Published: November 7, 2024

Open until: November 21, 2024, 16:00

Online Application via Email: sherbimetpublike.pr@rks-gov.net

Requests for clarifications are accepted until: November 16, 2024, 16:00