



TED UNIVERSITY

CMPE 313/SENG 214

Software Engineering

Citizen Portal Platform

Software Requirements Specification (SRS) Document

SECTION No-3 TEAM No-3

Submission Date

07/04/2024

Team Members: Mustafa Pınarcı, Ege İzmir, Bartu Özen, Ahmet Tokgöz, Onat Keser, Emir Can Tokalakoğlu

Revision History

Name	Date	Reason For Changes	Version

TABLE OF CONTENTS

1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Product Scope	1
1.4 References	1
2. Overall Description	1
2.1 Product Perspective	1
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints.....	4
2.6 Assumptions and Dependencies	4
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Hardware Interfaces.....	5
3.3 Software Interfaces	5
3.4 Communications Interfaces	6
4. System Features	6
4.1 Proposal Submission	6
4.2 Feedback and Voting	7
5. Nonfunctional Requirements	8
5.1 Performance Requirements.....	8
5.2 Safety Requirements.....	8
5.3 Security Requirements.....	9
5.4 Software Quality Attributes.....	9
6. Other Requirements	9
Appendix A: Glossary.....	10
Appendix B: Analysis Models.....	10

1. Introduction

1.1 Purpose

This SRS describes the functional and nonfunctional requirements for the initial release of the Citizen Portal Platform/ CPP for the municipality of the city. The purpose of this document is to serve as a guide for the project team responsible for the implementation and validation of the CPP. All requirements specified here are designated for initial release 1.0 unless otherwise stated. The intended audience includes developers, testers, and stakeholders involved in the development and deployment of the CPP.

1.2 Document Conventions

The functional requirements for this project are organized by use case within user class.

1.3 Product Scope

The CPP will permit citizens to vote on proposals, provide feedbacks for ongoing proposals and to offer new ones. For a detailed description of the features scheduled for full or partial implementation in this release, please consult the Citizen Portal Platform Vision and Scope Document.

1.4 References

1. TEAM, 3. Citizen Portal Platform Vision and Scope Document, www.citport.com/projects/PPP/PPP_Vision_and_Scope.docx
2. TEAM, 3. Cost Side City Counsel Standards, Version 1.3,
3. TEAM, 3. Government of X Cost Side City Ordinance, www.governmentx.com/ordinance/standards/city

2. Overall Description

2.1 Product Perspective

The Citizen Portal Platform is a new software system that replaces the current traditional processes for voting. The context diagram in Figure 1 illustrates the external entities and system interfaces for initial release. The system is expected to evolve after 2 years of the first release, ultimately connecting to the citizens and mayor in an ecosystem where proposals and ongoing processes are transparent to the citizens.

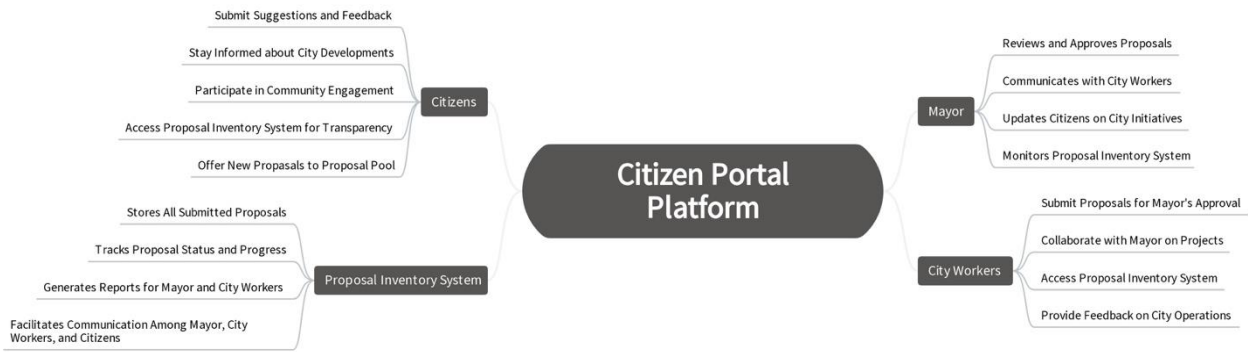


Figure 1. Context diagram for initial release of the Citizen Portal Platform.

2.2 Product Functions

Citizen Engagement: Permit citizens to participate in city governance by submitting suggestions and feedback, staying informed about city developments, engaging in community discussions, accessing the proposal inventory system for transparency, and contributing new proposals to the platform.

Mayoral Duties: The mayor shall be able to manage city affairs, including reviewing and approving proposals, communicating with city workers, updating citizens on city initiatives, and overseeing the proposal inventory system.

City Worker Collaboration: Support city workers in their collaboration with the mayor and by allowing them to submit proposals for approval, access the proposal inventory system, and provide feedback on city operations.

Proposal Management: Maintain a proposal inventory system to store all submitted proposals, track their status and progress, generate reports for the mayor and city workers, and facilitate communication among stakeholders.

2.3 User Classes and Characteristics

Mayor

Requires access to proposal review and approval functionalities, communication tools, and monitoring capabilities.

Citizens

Varied educational and technical backgrounds. Have an interest in civic affairs and a desire to engage with local government. Need interfaces for submitting suggestions, accessing information, and participating in community discussions.

City Workers

Employees of the city council or municipality. Involved in proposing and implementing projects, collaborating with the mayor, and providing feedback on city operations. May have varying levels of technical expertise.

User Class Importance

Mayor → Most important user class as they have decision-making authority and require comprehensive access to proposal management and communication functionalities.

Citizens → Important for fostering community engagement and transparency. Their input and participation are crucial for the success of the participatory governance platform.

City Workers → Vital for operational efficiency and effective collaboration with the mayor. They contribute to the proposal process and provide valuable feedback on city operations.

2.4 Operating Environment

- OE-1: The CPP shall operate correctly with the following web browsers: Windows Internet Explorer versions 7, 8, and 9; Firefox versions 12 through 26; Google Chrome (all versions); and Apple Safari versions 4.0 through 8.0.
- OE-2: The CPP shall operate on a server running the current corporate-approved versions of Red Hat Linux and Apache HTTP Server.
- OE-3: The CPP shall permit user access from the corporate Intranet, from a VPN Internet connection, and by Android, iOS, and Windows smartphones and tablets.

2.5 Design and Implementation Constraints

CO-1: The system's design, code, and maintenance documentation shall adhere to the municipality's software development standards and guidelines, as outlined in the Municipal Software Development Standard, Version 2.0.

CO-2: The system shall utilize the PostgreSQL database management system as per the municipality's database standard for enterprise applications.

CO-3: All frontend user interfaces shall comply with the latest web standards, including HTML5 and CSS3, to ensure cross-browser compatibility and accessibility.

CO-4: The system shall integrate with the municipality's existing authentication and authorization system for user management, following the protocols and standards specified by the municipality's IT department.

CO-5: The platform shall be designed to scale horizontally, following the municipality's scalability guidelines for enterprise systems.

CO-6: The system's architecture shall support internationalization and localization features to enable usage by citizens from diverse linguistic backgrounds, in accordance with the municipality's language support policy.

CO-7: The platform shall comply with relevant data privacy and security regulations, including but not limited to GDPR and CCPA, ensuring the protection of user data and privacy rights.

2.6 Assumptions and Dependencies

AS-1: The Citizen Portal Platform (CPP) operates to facilitate citizen voting processes, allowing users to submit proposals, provide feedback, and participate in voting activities at any time.

DE-1: The functionality of the CPP relies on integration with existing city council systems, including the Identification System, to validate users' identities and eligibility to participate in governance activities.

DE-2: The operation of the CPP is dependent on synchronization with the City Inventory System to accurate availability and tracking of proposals and feedback.

3. External Interface Requirements

3.1 User Interfaces

- UI-1: The user interfaces of the CPP shall adhere to the municipality's User Interface Design Guidelines, Version 1.0, guaranteeing consistency and usability across all screens and interactions.
- UI-2: The CPP shall include a help feature accessible from each page, providing contextual guidance on how to use the features and functionalities available on that specific page.
- UI-3: The web interface of the CPP shall support complete navigation and interaction using keyboard inputs alone, in addition to standard mouse and keyboard combinations, to accommodate users with accessibility needs and preferences.

3.2 Hardware Interfaces

- HI-1: The CPP communicates with hardware devices over standard communication protocols such as HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) for web-based interactions. Additionally, it may utilize other protocols such as TCP/IP (Transmission Control Protocol/Internet Protocol) for network communication between servers and client devices.

3.3 Software Interfaces

SI-1: Identification System

The CPP shall interface with the Identification System through a programmatic interface for the following operations:

- SI-1.1: To authenticate users and verify their identity before allowing access to the platform's functionalities.
- SI-1.2: To retrieve user profile information and permissions within the platform.
- SI-1.3: To update user data, such as city of residence and contact information, in the Identification System when modified within the CPP.

SI-2: Voting System

- SI-2.1: To retrieve active proposals available for voting from the Proposal Voting System and present them to citizens for consideration.
- SI-2.2: To record citizens' votes on proposals and transmit the voting data securely to the Proposal Voting System for tallying and analysis.
- SI-2.3: To query the Proposal Voting System to determine the status of proposals, including voting progress and outcome.

- SI-2.4: To synchronize proposal data between the CPP and the Proposal Voting System to ensure consistency and accuracy in proposal management and voting activities.
- SI-2.5: To handle proposal submission and withdrawal processes seamlessly between the CPP and the Proposal Voting System, maintaining data integrity and security throughout the process.

3.4 Communications Interfaces

- CI-1: The CPP shall send email notifications to citizens to confirm the acceptance of their proposal submissions, including details such as proposal title, submission date, and confirmation of receipt.
- CI-2: The CPP shall send email notifications to citizens to report any updates or changes to their proposals, including status changes (e.g., under review, approved, rejected), feedback received, and voting outcomes.

4. System Features

4.1 Proposal Submission

4.1.1 Description and Priority

This feature allows citizens to submit proposals for consideration by the city council. It is of high priority as it forms the core functionality of the Citizen Portal Platform (CPP), enabling citizen engagement in decision-making processes.

4.1.2 Stimulus/Response Sequences

User selects "Submit Proposal" option from the CPP dashboard.

System presents a form for the user to input proposal details.

User fills out the form with proposal title, description, category, and any supporting documents.

User submits the proposal.

System confirms receipt of the proposal and provides a unique reference number.

If any required fields are missing or contain invalid data, system displays error messages prompting the user to correct them.

4.1.3 Functional Requirements

REQ-1: The system shall provide a user interface for citizens to input proposal details, including title, description, category, and supporting documents.

REQ-2: The system shall validate proposal submissions to ensure all required fields are filled out and data is in the correct format.

REQ-3: The system shall assign a unique reference number to each submitted proposal for tracking purposes.

REQ-4: The system shall notify the user upon successful submission of the proposal, confirming receipt and providing the unique reference number.

REQ-5: If any errors occur during the submission process (e.g., network error, database error), the system shall display an error message to the user "Error occurred please retry the submission" and allow them to retry the submission.

4.2 Feedback and Voting

4.2.1 Description and Priority

This feature allows citizens to provide feedback on ongoing proposals and participate in the voting process. It is of high priority as it promotes citizen engagement and transparency in decision-making.

4.2.2 Stimulus/Response Sequences

User navigates to the list of ongoing proposals.

User selects a proposal to view details.

The system displays the proposal details along with options to provide feedback and vote.

User selects the "Provide Feedback" option.

System presents a form for the user to input feedback comments.

User submits the feedback.

System confirms receipt of the feedback.

User selects the "Vote" option.

System displays voting options (e.g., approve, reject, abstain).

Users select their voting preference.

System records the vote and updates the proposal status accordingly.

4.2.3 Functional Requirements

REQ-1: The system shall provide a user interface for citizens to view ongoing proposals and select individual proposals for feedback and voting.

REQ-2: The system shall allow users to submit feedback comments for each proposal, with an option to attach supporting documents if necessary.

REQ-3: The system shall validate feedback submissions to ensure all required fields are filled out and data is in the correct format.

REQ-4: The system shall notify the user upon successful submission of feedback, confirming receipt.

REQ-5: The system shall provide voting options for users to express their preference on each proposal, including options to approve, reject, or abstain.

REQ-6: The system shall record and tally votes for each proposal, updating the proposal status accordingly.

REQ-7: If any errors occur during the feedback or voting process, the system shall display an appropriate error message to the user “Error occurred please retry the page” and allow them to retry the voting and feedback page.

5. Nonfunctional Requirements

5.1 Performance Requirements

- The Citizen Portal Platform (CPP) shall respond to user interactions within 3 seconds under normal load conditions.
- The system shall process voting actions and feedback submissions with a response time of less than 5 seconds.
- The CPP should be capable of handling a minimum of 10,000 concurrent users without a significant degradation in performance.
- The system architecture should scale horizontally to accommodate increased user demand, with minimal impact on response times.
- The CPP shall be available for access by users 24/7, with a scheduled downtime of no more than 1 hour per month for maintenance purposes.
- The platform should be designed to withstand sudden spikes in user traffic without experiencing downtime or service interruptions.

5.2 Safety Requirements

- The system shall comply with GDPR and other relevant data protection regulations to safeguard user privacy rights and prevent unauthorized use or disclosure of personal information.
- User consent shall be obtained before collecting any personally identifiable information, and users shall have the option to delete their data from the system upon request.
- The CPP shall implement mechanisms to detect and prevent unauthorized modifications to system components, configurations, or data.
- Regular security audits and vulnerability assessments shall be conducted to identify and mitigate potential security risks and vulnerabilities in the platform.
- The system shall have a comprehensive disaster recovery plan in place to minimize data loss and downtime in the event of hardware failures, natural disasters, or cyber-attacks.
- Data backups shall be performed regularly and stored on different servers to ensure redundancy and resilience against catastrophic events such as data loss.

5.3 Security Requirements

- The platform shall implement strong authentication mechanisms, including multi-factor authentication, to verify the identity of users accessing the system.
- Passwords shall be securely hashed and stored using industry-standard cryptographic algorithms to prevent unauthorized access in case of data breaches.
- Access to system functionalities and data shall be restricted based on predefined roles and permissions assigned to users, with granular access controls enforced to prevent unauthorized actions or data exposure.
- All sensitive data, including personally identifiable information (PII) and authentication credentials, shall be encrypted both in transit and at rest using strong encryption algorithms to prevent unauthorized access or disclosure.
- All code changes shall undergo thorough code review and testing, including static code analysis and dynamic security testing, to identify and remediate security weaknesses before deployment to production environments.
- The platform shall maintain detailed audit logs of user activities and system events, including login attempts, access to sensitive data, and configuration changes, to facilitate forensic analysis, and incident investigation.

5.4 Software Quality Attributes

- **Adaptability:** The Platform shall be designed to adapt to changes in technology and user requirements, permitting updates within two weeks of identification.
- **Availability:** The Citizen Portal Platform shall aim for 99.9% operational time, excluding planned maintenance periods, to ensure continuous access for citizens and officials.
- **Correctness:** All functions of the CPP shall operate with an error margin of 0.01%.
- **Flexibility:** The system shall allow for the addition, modification, or removal of modules, ensuring impacts on existing functions are handled within no more than 30 days.
- **Interoperability:** The CPP shall ensure compatibility with existing municipal systems, committing to data exchange within a response latency not to exceed 3 seconds.

- **Maintainability:** The Code shall be documented to facilitate maintenance, with documentation updates to occur within a week of any change.
- **Portability:** The CPP shall be accessible across multiple platforms, including but not limited to Windows, macOS, iOS, Android, and web browsers.
- **Reliability:** The system shall maintain an error rate of less than 0.1% across all operations.
- **Reusability:** Components of the CPP shall be designed for reusability in future projects with a target of 50% reusability for the code base.
- **Robustness:** The CPP shall remain operational under unexpected conditions, with permitted downtime of no more than 0.5% outside planned maintenance.
- **Testability:** The system shall be designed to facilitate automatic testing, targeting a test coverage of 90%.
- **Usability:** The CPP shall be user-friendly, permitting an average learning time of under an hour for citizens, with 80% of users navigating the system independently post-learning session.

6. Other Requirements

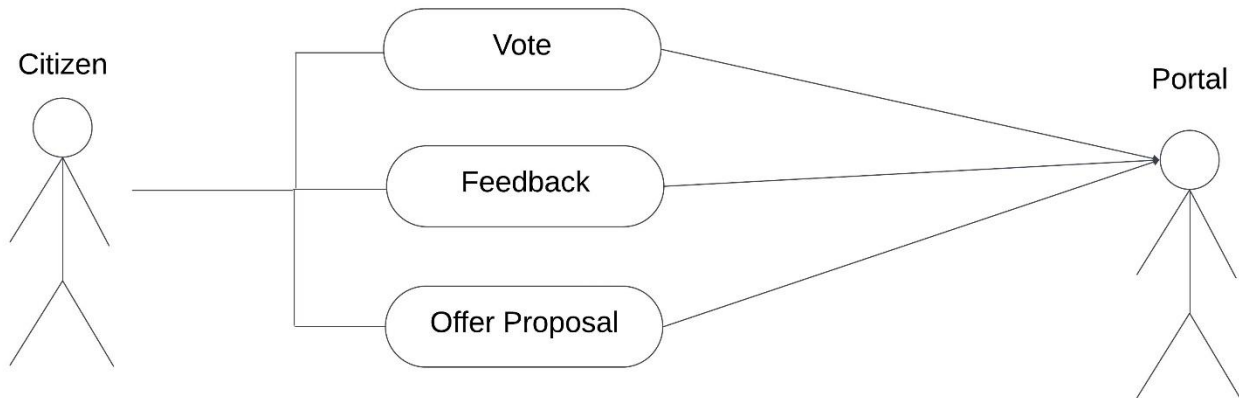
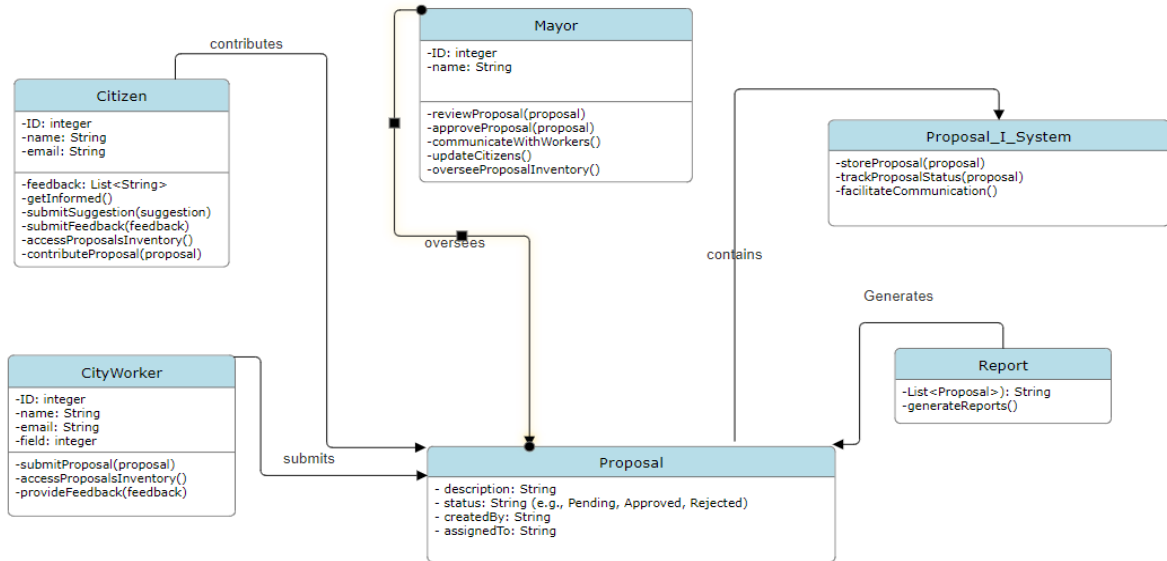
- **Database Requirements:** The CPP shall utilize SQL, supporting transactions of at least 10,000 records per second without degradation in performance.
- **Internationalization Requirements:** The CPP shall offer multilingual support, starting with English, Spanish, and additional languages.
- **Legal Requirements:** The CPP shall follow the General Data Protection Regulation (GDPR) and other applicable international or national data protection laws.
- **Accessibility:** The CPP shall be designed to the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA standards to ensure accessibility for users with disabilities.

Appendix A: Glossary

- **Citizen Portal Platform (CPP):** A system designed to engage citizens in governance by enabling voting, feedback, and proposal submission.
- **GDPR (General Data Protection Regulation):** A regulation in EU law on data protection and privacy.
- **WCAG 2.1 (Web Content Accessibility Guidelines 2.1):** Recommendations for making web content accessible to people with disabilities.
- **SQL (Structured Query Language):** A standardized programming language for managing relational databases.
- **Red Hat Linux:** A widely-used Linux operating system distribution known for its stability.
- **Apache HTTP Server:** An open-source web server software.
- **PostgreSQL:** An open-source relational database management system.
- **HTTP (Hypertext Transfer Protocol):** A protocol used for transmitting web pages.
- **HTTPS (HTTP Secure):** A secure version of HTTP.
- **TCP/IP (Transmission Control Protocol/Internet Protocol):** A set of protocols for the Internet.
- **VPN (Virtual Private Network):** A service that protects internet communication.
- **API (Application Programming Interface):** A set of rules enabling software applications to communicate with each other.

Appendix B: Analysis Models

Class Diagram



ID and Name:	1-Vote
Actors:	Citizen and Portal.
Description:	Citizen logs in to Portal to vote.
Trigger:	Citizen displays the candidates and selects one then clicks vote button.
Preconditions:	Citizen must be a permanent resident of the city and must be logged in to the system.

Postconditions:	Citizen votes.
Normal Flow:	Citizen enters his information and logs in to the portal. Then, votes and exits the system.
Alternative Flows:	
Exceptions:	Citizens may select more than one option. In that case, the system displays a warning and if the citizen continues, that vote becomes invalid.
Non functional requirements	The system checks the information of citizens while logging in due to Security requirements. It also sets a deadline for reliability requirement.

ID and Name	2-Feedback
Actors	Citizen and Portal.
Description	Citizen monitors the proposals of another citizen and gives them feedback.
Trigger	Citizen logs in to system and clicks Provide Feedback. Then, selects a ongoing proposal and writes feedback.
Preconditions	Citizen must be a permanent resident of the city and must be logged in to the system.
Postconditions	Citizen successfully gives feedback.
Normal Flow	Citizen enters his information and logs in to the portal. Then, does the feedback operation and exits the system
Alternative Flow	Citizens may select more than one proposal. In that case, he must write each one of them separately.
Exceptions	Citizen may try to select a proposal that does not exist. In that case, System displays an error message.
Non functional requirements	The system does not show the origin of an ongoing proposal due to Security requirement of citizens. The System limits the feedback operation to a specific amount due to Interoperability requirement.

ID and Name	3- Offer Proposal
Actors	Citizen and Portal
Description	Citizen offers proposal for City.
Trigger	Citizen logs in to system and clicks Offer Proposal button. Then writes the proposal with details and submits it.
Precondition	Citizen must be a permanent resident of the city and must be logged in to the system.
Postcondition	Citizen proposes.
Normal Flow	Citizen enters to system then does the offering proposal and exits the system.
Alternative Flow	Citizens may try to offer more than one proposal. In that case, he must write each one of them separately.
Exceptions	Citizens may try to offer unsafe proposals for residents. In that case, System does not accept it and it does not put it to process
Non functional requirements	The system checks if the proposal is hazardous or not by AI, due to Security requirement