
DPEX Package Tracking System

System Requirements Specification

Version 1.0.0

November 26, 2023

Table of Contents	i
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Product scope	1
1.3 Glossary	1
1.4 References	1
1.5 Overview	1
2. General Description	1
2.1 Product Perspective	2
2.1.1 System interfaces	2
2.1.2 User interfaces	2
2.1.3 Hardware interfaces	2
2.1.4 Software interfaces	2
2.2 Product Functions	2
2.3 User Characteristics – Sheyla Lamar	3
2.4 Constraints	3
2.5 Assumptions and Dependencies	3
2.6 Apportioning of Requirements	4
3. Specific Requirements	4
3.1 External Interfaces	4
3.2 Functions	4
3.2.1 Use Case: Create New User	4
3.2.2 Use Case: Assign Roles and Permissions	4
3.2.3 Use Case: Generate Monitoring Reports	5
3.2.4 Use Case: System Performance Statistics	5
3.2.5 Use Case: Register Electoral Package	5
3.2.6 Use Case: Manage Electoral Package Information	5
3.2.7 Use Case: Create Checklist for Electoral Kit	6
3.2.8 Use Case: Generate Tracking Code for Electoral Package	6
3.2.9 Use Case: Package Status Change Alerts	6
3.2.10 Use Case: Authorized Access to Sensitive Information	6
3.2.11 Use Case: Security Audits	7
3.2.12 Use Case: Automatic Data Verification	7
3.2.13 Use Case: Manual Data Verification	7
3.2.14 Use Case: Secure Client-Server Communication	7
3.2.15 System Functional Requirements	8
3.2.4 Nonfunctional Requirements	9
3.2.5 Deleted Requirements	9
4. Appendixes	11

Table of Contents

[illegible]

1. Introduction

1.1 Purpose

This document serves three main objectives:

- To explain the methodology used to analyze and enhance the initial requirements provided by DPEX.
- To state the requirements for the DPEX Package Tracking System.
- To present a detailed analysis of the requirements, including dependencies, resolved issues, comprehension of the requirements, and their interrelations.

As such, this document goes beyond a mere requirements specification; it also serves as an articulation of the derivation process from the input and the interconnected nature of the requirements

1.2 Product scope

The product is a software system that is divided into 2 parts: tracking of electoral packages abroad and management of these within the DPEX.

1.3 Glossary

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document:

DPEX: Acronym for Dirección de Procesos en el Exterior (Foreign Processes Management Office).

JRE: Acronym for Java Runtime Environment

JDK: Acronym for Java Development Kit

1.4 References

- IEEE 830-1998

1.5 Overview

This document is based on the IEEE830-1998 specification.

The following information can be found in this document:

- Section 2: General description of product, including user dependency analysis and enterprise requirements.
- Section 3: Functional & Non-functional requirements.
- Section 4 describes the requirements analysis process.

2. General Description

The system aims to be a solution to allow the DPEX members to know the information of the electoral packages sent abroad, destination and tracking during the electoral process. Once finished, the system will focus on managing the packages for storage or possible counting.

2.1 Product Perspective

The DPEX Package Tracking System is a self-contained system. It will run on the user's command line interface and dynamic data sources such as databases will be stored in the cloud; this means the user will only need to have access to a computer or mobile device running Windows, MacOS or Linux. All the system features are all built into itself, and no other software or hardware device is required to function.

2.1.1 System interfaces

As stated in section 2.1 the system is self-contained, relying on very little in external software interfaces. However, the system will require interfaces with the installed computer's hardware. The system is to be a web-enabled system, meaning that all user interaction it is required to have an internet connection to retrieve and modify data. The System interfaces required on the system server are the following:

- Network interface to a network with an internet connection
- Database connection to the MongoDB database containing data, such as user authorization information, package tracking information, etc.

2.1.2 User interfaces

All user interfaces occur through the device's command-line.

2.1.3 Hardware interfaces

There are no hardware interfaces to this system.

2.1.4 Software interfaces

On the user end, it is required to have JRE installed.

2.2 Product Functions

The Enterprise functional & non-functional requirements from DPEX have been analyzed and all issues and ambiguities resolved. This document is for review by the customer (DPEX), and the changes to the spec should be analyzed for acceptability to DPEX.

The general outline of the product functions is as follows:

- Registration and information management
- Electoral kits preparation
- Shipment and tracking
- Secure handling
- Data verification

- Technology
- Communication

2.3 User Characteristics – Sheyla Lamar

The system is divided into two types of users: administrators and employees. Administrators can update information regarding consular offices, population and other sensitive data. Employees can register and mark packages as well as counting the votes.

User type	Administrator
Role	Supervisor
Skills	Communication, time management, leadership, experience in the area
Activities	Supervision of the group of electoral administrative assistants, inside the DPEX.

All users must have knowledge of how to work with command-line programs. Since all interactions with the system is the computer command-line, the system cannot be used without access and knowledge of it.

2.4 Constraints

There are several constraints which the system must abide by during development. The system must be developed within their limits. These constraints dictate a few of the functional and nonfunctional requirements specified by this document. Others are because of a requirement specified to us by our customer. All are important to be aware of during the implementation of the software system.

- System is to be developed for use as a command-line application. This will limit the ability to create user-friendly graphical interfaces.
- System is to be developed in Java.
- Data must be stored in a remote database.
- Passwords must be sent and stored in encrypted form.
- Not all users are able to access and/or modify the different sets of data.
- System must be robust enough to handle many concurrent queries.
- Server-Client communication must be done over TCP connections

2.5 Assumptions and Dependencies

- Backend system will run on any of the previously listed operating systems: Linux, MacOS or Windows, will have to be able to run MongoDB 7.0 or newer
- The program will run on any if the previously stated OSes and they must be able to run JRE 17 or newer. This means Java is required to be installed on the system to work.

- The system is supposed to run on a secure environment, meaning that, for example, all connections should be made through VPNs or any other measures, that should be taken on the system administrator side as they are not part of the system functionality.

2.6 Apportioning of Requirements

The system should be fully fledged and operational since the very first release, as the electoral process can only be held once in a couple of years. However, there can be some non-critical features that could be developed in the future such as:

- i) Automated initial counting: the system could implement an OCR algorithm that counts (and recounts) the votes automatically, this would help to speed up the process.
- ii) Graphical user interface: if the command-line program is found to be unfriendly or hard to use for the end users, we may need to considerate developing a graphical user interface to improve ease of use.

3. Specific Requirements

3.1 External Interfaces

(esta parte lo dejo en blanco de momento, necesitamos al menos un bosquejo del diseño del proyecto, aunque el Edison dijo que todo seria por consola asi que talvez no necesitemos esta parte)

3.2 Functions

System functional requirements are specified by use cases and specific requirements. The use case helps understand system behavior, and the specific requirements extend the information from the use case.

3.2.1 Use Case: Create New User

Use Case Description:

Administrators will be able to create new user accounts to assign roles and permissions.

Specific requirements:

- The system will provide an interactive form for entering electoral package information.
- Real-time validations will be performed to ensure data integrity and consistency.
- Users will be able to attach documents related to the package, such as shipping records.

3.2.2 Use Case: Assign Roles and Permissions

Use Case Description:

Administrators will be able to assign specific roles and define permissions for each user.

Specific requirements:

- Available roles will include roles such as administrator, employee, and supervisor, each with specific sets of permissions.
- Changes to roles will require authorization from another administrator to prevent potential abuse of privileges.

3.2.3 Use Case: Generate Monitoring Reports

Use Case Description:

The system will generate detailed reports on the status and location of election packages.

Specific requirements:

- Reports will be generated using specific commands from the command line interface.
- Reports will include information such as shipping dates, current locations, status changes, and details of any issues or issues encountered during the election process.
- Reports will be available in exportable formats for easy review and presentation.

3.2.4 Use Case: System Performance Statistics

Use Case Description:

Statistics on system performance will be generated to evaluate efficiency and detect possible improvements.

Specific requirements:

- Performance metrics such as response times and resource utilization will be recorded.
- Administrators will be able to access detailed statistics reports.

3.2.5 Use Case: Register Electoral Package

Use Case Description:

The user enters the detailed information of an electoral package and registers it in the system.

Specific requirements:

- The system will provide an interactive form for entering electoral package information.
- Validations will be carried out to guarantee the integrity of the data entered.
- Authorized users will be able to access and modify the registered information.

3.2.6 Use Case: Manage Electoral Package Information

Use Case Description:

Authorized users will be able to manage the information of the registered electoral packages, including updating and deleting records.

Specific requirements:

- Functions will be provided for updating and deleting election package records.
- Modification of information will be subject to user authentication.

3.2.7 Use Case: Create Checklist for Electoral Kit

Use Case Description:

Authorized users can create checklists to ensure the inclusion of necessary items in election kits.

Specific requirements:

- The system will allow the creation of custom checklists.
- Checklists will be associated with specific types of election kits.

3.2.8 Use Case: Generate Tracking Code for Electoral Package

Use Case Description:

The system will generate unique tracking codes for each election package sent abroad.

Specific requirements:

- Tracking codes will be unique and used to track the status and location of packages.
- Users will be able to access tracking information through the command line interface.

3.2.9 Use Case: Package Status Change Alerts

Use Case Description:

The system will send alerts to users when there are significant changes in the status of an election package.

Specific requirements:

- An alert system will be implemented to notify users of relevant changes.
- Alerts will be sent in a timely and efficient manner.

3.2.10 Use Case: Authorized Access to Sensitive Information

Use Case Description:

Only authorized users will be able to access and manipulate sensitive information about electoral packages.

Specific requirements:

- Authentication measures will be implemented to ensure authorized access.
- The manipulation of sensitive information will be restricted to specific roles.

3.2.11 Use Case: Security Audits

Use Case Description:

The system will perform periodic audits to detect tampering attempts or unauthorized access.

Specific requirements:

- The system will generate audit logs that will include security-related activities.
- Administrators will be able to review audit reports.

3.2.12 Use Case: Automatic Data Verification

Use Case Description:

The system will perform automatic data checks to identify possible inconsistencies or errors.

Specific requirements:

- An automatic mechanism will be implemented to verify the accuracy of the data.
- Data verification reports will be generated for review.

3.2.13 Use Case: Manual Data Verification

Use Case Description:

Authorized users may perform manual verifications to confirm the accuracy of the stored information.

Specific requirements:

- Tools will be provided for users to perform manual verifications.
- The results of manual checks will be recorded for future reference.

3.2.14 Use Case: Secure Client-Server Communication

Use Case Description:

The system will establish secure connections between the client and the server via TCP to ensure the security of communication.

Specific requirements:

- Security protocols will be implemented for data transmission.
- Secure communication will be a priority in unstable network environments.

3.2.15 System Functional Requirements

3.2.3.1) Database management

The system will have an external database containing information about all potential voters. This information should consider details such as the country of residence, the type of division to which they belong, district, and consular office.

3.2.3.1.1) Package tracking

The system must have the capability to track packages sent abroad before the electoral process. For this purpose, the system should assign a unique identification (ID) to each package, enabling its monitoring from the CNE offices to its respective assigned consular office.

3.2.3.1.2) Package classification

The system must categorize each consular office into three distinct categories, which will depend on the number of registered voters in each location.

3.2.3.2) Kit preparation

The system must generate guidelines for the preparation of each electoral kit, which may vary according to the requirements of each consular office.

3.2.3.2.1) Data Verification:

The system must verify the integrity of the data once the electoral kits return to the central offices of the CNE. In the event of disruptions or disputes, appropriate actions will be taken.

3.2.3.3) Secure handling

Due to the fact that the electoral packages will return to the CNE offices, transparency and security must be ensured in the transportation, packaging, and delivery of the packages to and from the consular offices and the CNE.

3.2.4 Nonfunctional Requirements

3.2.4.1) Usability

Due to the nature of the information handled during the electoral process, the system should be as accessible and intuitive as possible to minimize human errors.

3.2.4.2) Security

Credentials should be created so that only authorized individuals can access the system and perform their respective tasks.

3.2.4.3) Performance

The system should deliver an adequate level of performance to avoid delays in program response times or potential performance degradation during the shipping process

3.2.4.4) Extensibility (Scalability?)

The system must be extensible and scalable to handle large volumes of data during the process of transferring electoral packages.

3.2.4.5) System logs(Backups?)

Backups of the system's information must be generated periodically to prevent the loss of data caused by potential failures of the system itself or other types of errors.

3.2.4.6) Communication:

Continuous communication will be maintained with consular offices to address any issues or updates related to electoral packages.

3.2.4.7) Fault tolerance

The system must be designed in such a way that potential errors do not compromise the integrity of the entire system to prevent delays during the shipping process or loss of information.

3.2.5 Deleted Requirements

There are no deleted requirements as of this revision of the document.

4. Appendixes

Within the aspect of the program, it was necessary to know a little more about how the electoral packages work, some of their components and the distribution of the content during the voting period:

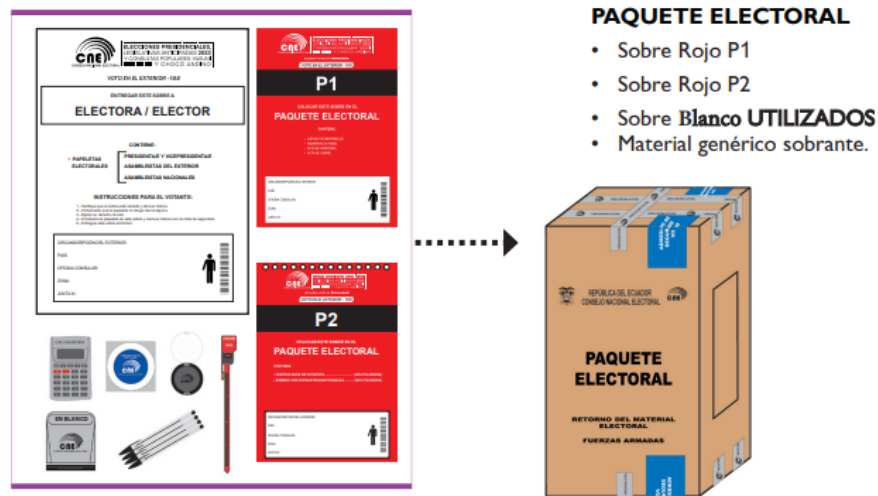


Figure 1. electoral packages distribution

In this way, the Electoral Kits are generated, these are packages which have the following:

- Screen and urn
- Electoral register.
- Voting Certificates
- Ballot papers
- Generic materials.
- Minutes of opening, installation and scrutiny.
- Envelopes “P1”, “P2”, “T1”.

In the same way, a laptop and a scanner are attached, useful for accessing the CNE system and uploading data necessary for the initial count of the votes, these are separated from these packages. It should be noted that not all places have a large concentration of Ecuadorians, so they are divided as follows:

- "cne": The "cne" collects places that register between 1 to 100 people, these electoral packages are sent to the consular offices which will be the polling place, among these sites we have Canberra, Australia; Tel Aviv, Israel. There is only one electoral package in this one
- "Mixto": The "Mixto" collects between 101 and 889 people, where the consuls determine a place to vote. Among these places we find: Buenos Aires, Argentina; New York—Hudson Valley; United States.
- "Gender": Places that register 890 people or more are collected, in this way the votes are made in places of large concentrations, where they are divided into male and female boards, in this way they receive more packages compared to the other divisions. These sites include Madrid, Spain; Milan, Italy.

And in the same way, making known what the most important documents are within these packages, except for the voting ballots.



Figure 2. T1 and P1 proceedings.

Once the necessary knowledge about the packages has been obtained, we can better understand how the program could be delivered to our client. The image shows the classes and different objects that can be within the program, thus generating version 2.0.0 of our classes diagram.

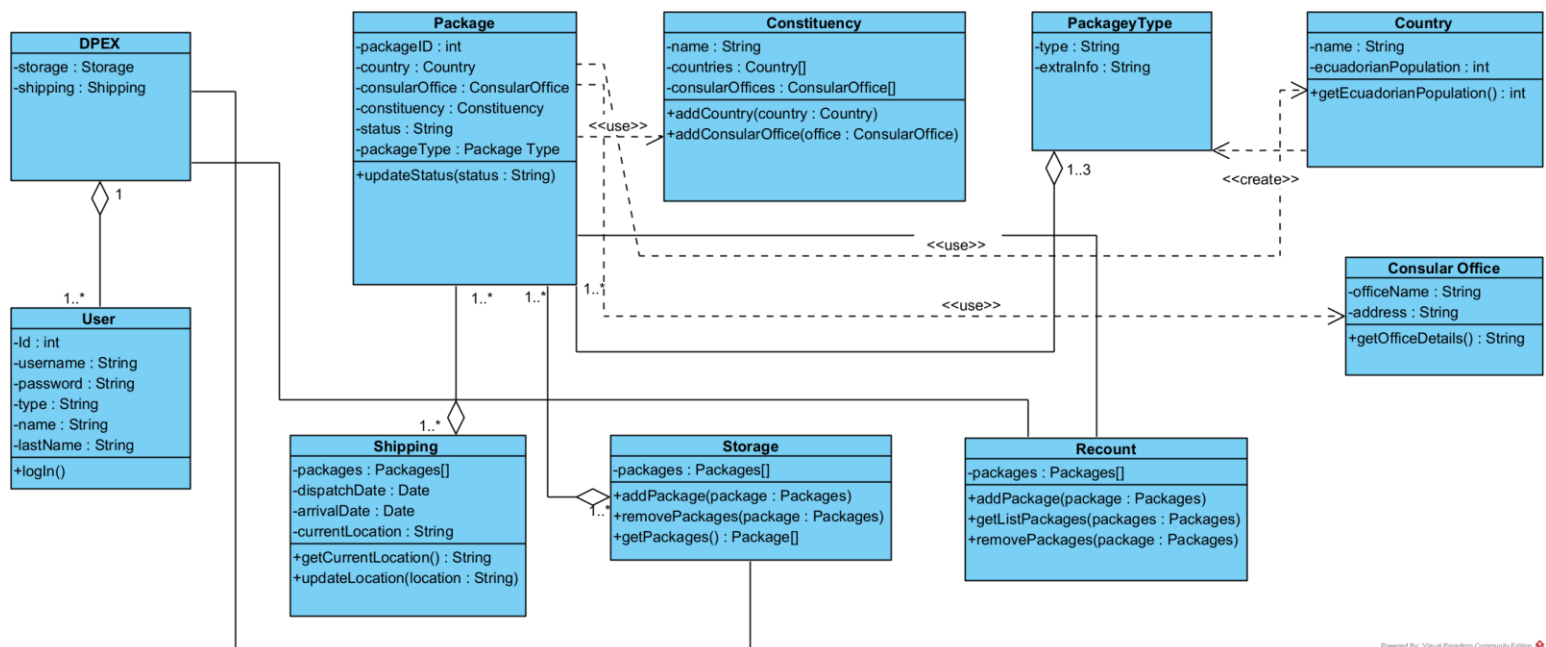


Figure 3. Classes Diagram for the system

