Software Requirements Specification

for

Technical Support Application

Version 3.0 approved

Prepared by

Pérez Morales Owen
Luis Alberto Gutiérrez Palomares
Mejia Gonzalez David
Cristopher Anahel Gonzalez Leyva

18/01/2025

Document information

Date	Review	Author	Verified by quality control.
Monday, January 18 2025	**Review 1**	Pérez Morales Owen Luis Alberto Gutiérrez Palomares Mejia Gonzalez David Cristopher Anahel Gonzalez Leyva	Marylu Lara Lagunes

By the client	By the Supplier Company
Marylu Lara Lagunes	Pérez Morales Owen Luis Alberto Gutiérrez Palomares Mejia Gonzalez David Cristopher Anahel Gonzalez Leyva

Table of Contents

1. Introduction

1.1 Purpose

This document explains those software requirements that are responsible for managing and facilitating consumer support. The system will allow companies with the process to know exactly the problem that a client has with the service offered by said company, the system will offer different options for the user such as registering problems, scheduling technical visits, receiving remote help and carrying out track the status of the request. The document focuses on the main parts of the system.

1.2 Document Conventions

This SRS uses simple text formatting to ensure easy readability. The following conventions are used:

- **Bold text** is used to highlight key terms or sections.
- *Italics* are used for examples or notes.
- All requirements are listed in normal text and are numbered for easy reference.
- Each requirement has its own priority, meaning the priority level is specified individually for every requirement.

No complex typographical conventions or special symbols are used in this document to keep it simple and clear.

1.3 Intended Audience and Reading Suggestions

This document is intended for several types of readers:

- **Developers:** To understand the functional and technical requirements needed to build the system.
- **Project Managers:** To track project scope and ensure all requirements are met.
- **Testers:** To use the requirements for creating test cases and validating the system for future uses.
- **Documentation Writers:** To prepare user manuals based on the system's

features.

The document is organized into sections covering an overview of the system, followed by detailed functional and non-functional requirements. Readers should start with the overview sections, those like the Purpose and Product Scope, before moving into the more specific requirements.

1.4 Product Scope

We will develop a technical support app called SupportTech. This software will help the company to manage problems related to a product or service that customers present, in addition to keeping track of the problems presented. Modules include managing users by role, tracking issues, whether they are resolved or not, issue history, and feedback. The software is made to improve information management in this area.

1.5 References

- *IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications*

2. Overall Description

2.1 Product Perspective

The technical support software is an app for the management and monitoring of technical support work carried out for clients who have such problems with their products or services. It is for companies that need to improve the way they manage information from the technical support area, that need to improve the way in which they receive help requests and monitor their progress, in addition to generating reports on the work done. The system provides full control through different modules that include the management of employees, clients, requests, reports, to ensure greater efficiency.

2.2 Product Functions

- **Employee and User Management:** Register employees, assign roles, and manage permissions based on hierarchy.
- **Roles and Permissions Management**: User roles define their permissions in the system, like creating tasks, teams, or reports.
- **Request Management:** Manage requests made by customers.
- **Analysis and reports:** Statistics of requests made by clients, in addition to reporting the problems presented.
- **Chat Support:** Support a customer chat with an employee to resolve issues remotely.

2.3 User classes and characteristics

User type	Team Member	Team Leader
Formation	Technician	Technician
Skills	App development, know node.js and javascript programming languages, MySQL, knowledge of APIs	App development, know node.js and javascript programming languages, MySQL, knowledge of APIs
Activities	Perform tasks.	Create tasks and manage the team.

2.4 Operating Environment

The platform will run in a web-based environment. It will be compatible with modern hardware and software systems, specifically:

- **Hardware Platform:** Standard web server configurations, such as Intel-based servers with 8GB+ RAM and sufficient storage for user data.
- **Operating System:** Linux (Ubuntu) or Windows 8+
- **Database:** MySQL for storing data related to the users and companies.
- **Web Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge (latest versions).

2.5 Design and Implementation Constraints

- The application can only be accessed through a login and authentication system.
- The permissions structure must be strict so that only authorized users can access certain parts of the application.
- The data storage system must have security protocols.
- The performance of the application must be the most optimal so that mobile devices can run the application without problems.

Minimum Requirements:

- **Operating System:** Windows Server 2016 or Linux (Ubuntu 18.04+)
- **Processor:** Dual-core Intel or AMD, 2.5 GHz or faster
- **Memory:** 4 GB RAM
- **Storage:** 10 GB available disk space
- **Browser Support:** Chrome (latest), Firefox (latest), Safari 13+

Recommended Requirements:

- **Operating System:** Windows Server 2019 or Linux (Ubuntu 20.04+)
- **Processor:** Quad-core Intel or AMD, 3.0 GHz or faster
- **Memory:** 8 GB RAM
- **Storage:** 20 GB available disk space
- **Browser Support:** Chrome (latest), Firefox (latest), Safari (latest), Microsoft Edge

2.6 User documentation

- User Manual: Guides to know how to use the system, explaining how an employee can make reports, view problem requests, talk to the customer. In addition to having a section for clients and how they can use the application to be able to upload their problems to the application.
- Administration Manual: Guide for application administrators to learn how to use the application, create roles, add new users.

2.7 Assumptions and Dependencies

- Users are expected to have knowledge about technologies and how to use a mobile application.
- Users are expected to have internet access to be able to use the application correctly.
- The application will be dependent on the database and API services being available and operating throughout the life cycle of the application.

2.8 Possible Future System Evolution

The application will have the possibility of evolving into a better application with new implementations that will make the app more complete.

- Advanced tools for data analysis: being able to perform better analyzes of the technical problems that have arisen.
- Integration of video calls: Integrate the possibility of video calls in the application to be able to carry out better communication with the client.

3. External Interface Integration

3.1 User Interfaces

3.1.1 Technologies for the Interface

The technologies that will be used for the user interface in the app will be javascript and css. Javascript to create the structure of the application and CSS to give styles to what is being done.

3.2 Hardware interfaces

The software will be made to be executed on different mobile devices with different components for a better experience and so that the app can be executed on as many devices as possible. The hardware interface includes input devices and output devices, in addition to network connections, storage. Each of these interfaces and the configuration that each one must have will be shown in more detail below.

3.2.1 Input Device

For input devices, what is needed is the touch screen of the mobile device.

- **Configuration**: Touch keyboard with qwerty and numeric language.
- **Compatibility**: Android operating system.

3.2.1 Output Device

For output devices what is needed is the screen of the mobile device.

- **Configuration**: Resolution configuration and color calibration.
- **Compatibility**: It is compatible with multiple screens of different mobile devices.

3.3 Software Interfaces

3.3.1 Software Compatibility

The system will be made to be compatible with different versions of the Android operating system, Android version 10.0 and higher, as long as the device's processor is compatible with the Android version.

3.3.2 Minimum Processor Requirements

- **CPU**: Quad-core 1.5 GHz or higher (64-bit recommended).

3.3.3 Examples of Compatible Processors

- Qualcomm Snapdragon 439.
- MediaTek Helio G35.
- Samsung Exynos 850.

3.3.4 Additional Considerations

- **Performance**: Even though these processors are old and basic, they can still run office applications and small programs without major problems.

3.3.5 MySQL Relational Database for Data Storage

MySQL will store and manage program data, including users, teams, projects, and tasks, using tables like task, subtask, employee, users, department, area, team, planning, progress, etc. MySQL will allow performing queries to select, insert, update, and delete data, as well as using subqueries, indexes, and views.

3.3.6 Database Connection

The connection to the database will be made through node.js, which will help us implement MySQL technologies with the project.

4. Internal requirements

Number of requirements	**REQ-001**		
Requirement name	User Management		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-002**		
Requirement name	Ticket Management		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-003**		
Requirement name	Location Management		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-004**		
Requirement name	Equipment Management		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-005**		
Requirement name	Maintenance Management		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-006**		
Requirement name	Reports		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-007**		
Requirement name	Push Notifications		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-008**		
Requirement name	Feedback		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-009**		
Requirement name	Performance Requirements		
Type	Requirement	Restriction	
Fuente del requisito	Specification document		
Source of requirement	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-010**		
Requirement name	Security Requirements		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-011**		
Requirement name	Safety Requirements		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-012**		
Requirement name	Availability		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-013**		
Requirement name	Maintainability		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-014**		
Requirement name	Portability		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

Number of requirements	**REQ-015**		
Requirement name	Usability		
Type	Requirement	Restriction	
Source of requirement	Specification document		
Requirement Priority	High/Essential	Medium/Desired	Low/Optional

4.1 Functional Requirements

4.1.1 Functional Requirements: User Management Module

This module handles the registration and management of users within the system. Users are assigned specific roles (Admin, Technician, Client), which determine their access and permissions. The module allows users to update their profiles and reset passwords. It ensures that the right individuals have access to the appropriate features, enabling efficient task assignment and secure system management.

4.1.2 Functional Requirements: Ticket Management Module

This module is responsible for managing the lifecycle of support tickets. Clients can create tickets detailing their issues, which are then assigned to technicians. Technicians can update the status of tickets (example, "In Progress", "Resolved") and add notes. This module ensures that all support requests are tracked, processed efficiently, and resolved in a timely manner, maintaining clear communication between clients and technicians.

4.1.3 Functional Requirements: Location Management Module

The Location Management Module stores and manages information about the company's locations. Administrators can register new locations, including details such as name, address, and contact information. This module helps organize the physical assets and assigns technicians to specific locations, ensuring that support and maintenance tasks are carried out where needed.

4.1.4 Functional Requirements: Equipment Management Module

This module is used to register and manage equipment at each location. Each piece of equipment is tracked with key information like model, serial number, and maintenance history. It allows the administrators to schedule maintenance and keep an updated record of all equipment, ensuring the efficient operation of physical assets across all locations.

4.1.5 Functional Requirements: Maintenance Management Module

The Maintenance Management Module tracks all scheduled and completed maintenance tasks for equipment. Administrators can schedule regular maintenance for equipment and assign technicians to carry out the tasks. Technicians can log details of completed work. This module helps ensure that all equipment is regularly serviced and that maintenance is documented for future reference.

4.1.6 Functional Requirements: Reports Module

This module generates reports related to support tickets, maintenance tasks, and technician performance. Administrators can filter and export these reports in formats such as PDF or Excel, allowing for deeper analysis of system usage and performance. It provides key insights to improve operations and identify areas for improvement.

4.1.7 Functional Requirements: Push Notifications Module

The Push Notifications Module is responsible for sending real-time alerts to users about ticket updates, maintenance schedules, and other important events. Users can customize their notification preferences, choosing what types of notifications they wish to receive. This module helps keep all users informed and ensures timely responses to important system updates.

4.1.8 Functional Requirements: Feedback Module

This module collects feedback from clients after ticket resolution or maintenance tasks. Clients can rate the service and provide comments, which are then analyzed by administrators. The feedback helps to measure customer satisfaction and identify opportunities for improving the support service.

4.2 Nonfunctional Requirements

4.2.1 Performance Requirements

The system must operate efficiently, handling a high number of simultaneous users without delays. For example, loading times for tickets and equipment lists should be minimal, and actions like updating ticket statuses or viewing equipment details must be processed quickly, ensuring smooth user interactions without noticeable lag.

4.2.2 Security Requirements

Security measures protect the system from unauthorized access and data breaches. This includes secure login with strong password policies, encrypted data transfer, and restricting sensitive actions to specific roles. For instance, only admins should have access to user management, while technicians and clients have limited functionalities based on their roles.

4.2.3 Safety Requirements

The system should function reliably and error-free during continuous use. For example, in case of a network issue, users should be able to resume their tasks without data loss, and actions like creating or updating tickets should still be processed even if there’s a temporary interruption. The system must ensure high reliability and avoid disruptions in service.

4.2.4 Availability

The system must be highly available, ideally operating 99.9% of the time. This ensures minimal downtime for maintenance or issues. Regular checks and updates should be performed to avoid service interruptions, and the system should be designed to quickly recover from any failures.

4.2.5 Maintainability

The system should be easy to maintain and update. This includes clear code documentation and a straightforward process for fixing bugs or adding features. Maintenance tasks should be carried out efficiently by the development team, and administrators should easily handle tasks like updating user information or managing equipment data.

4.2.6 Portability

Portability refers to the system's ability to run on both iOS and Android without requiring significant modifications. Using a cross-platform framework like React Native, the same codebase should function seamlessly on both platforms, ensuring consistency across devices while minimizing platform-specific code.

4.2.7 Usability

The system should be easy to use, with a user-friendly interface that requires minimal training. It should be intuitive, with clear navigation and helpful features like progress bars and notifications. The app must be responsive, providing a smooth experience whether on a smartphone or tablet, and adapting to different screen sizes.

5. Other Requirements

The project must be completed and delivered by April 10th. All team members are expected to finalize their tasks and ensure that all requirements are met before this deadline. This timeline is critical for the project's success, and any potential delays should be communicated promptly to ensure timely resolution.

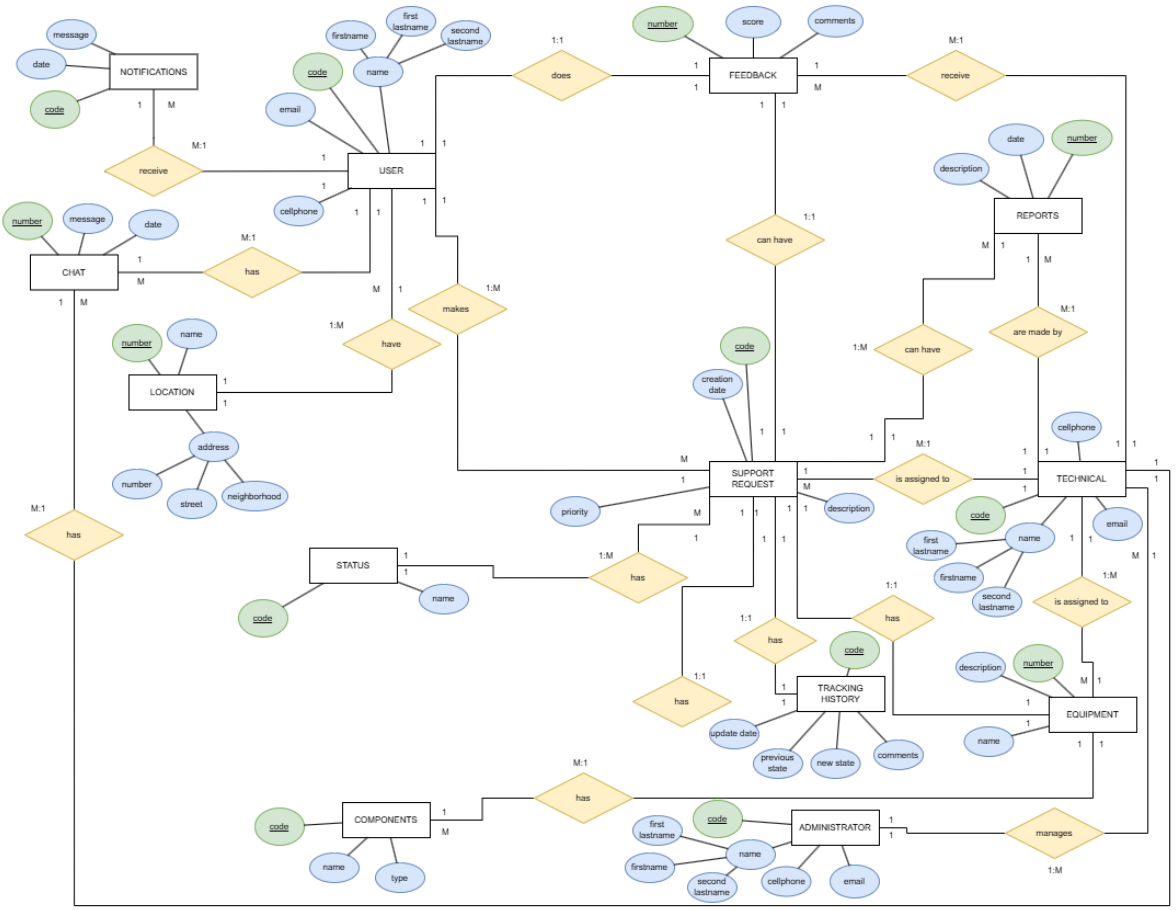
6 . Appendix

6.1 Appendix A: Glossary

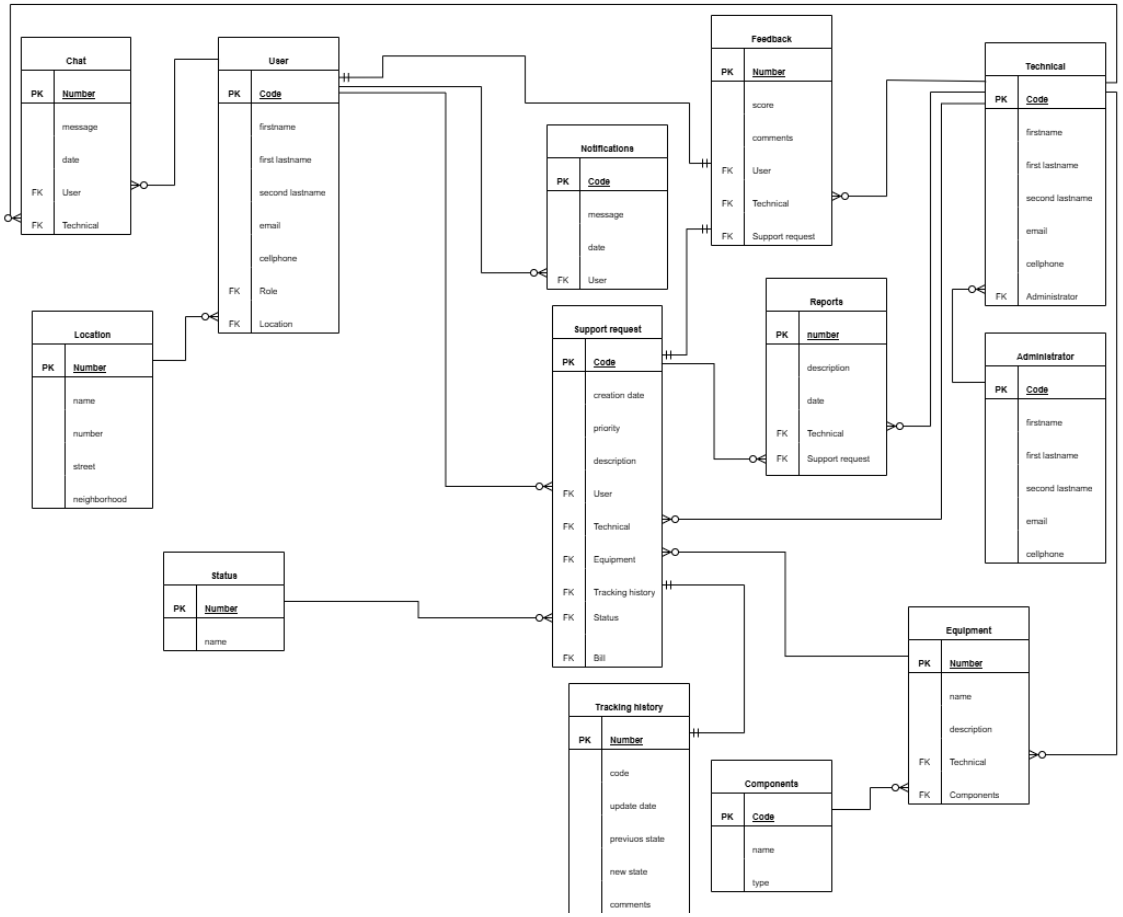
Introduction
This appendix provides a glossary of terms used in the document to help clarify technical and specific language for this project.

6.2 Appendix B: Analysis Models

6.2.1 ERD diagram



6.2.2 RM


Chat
PK Number
message
date
FK User
FK Technical
User
PK Code
firstname
first lastname
second lastname
email
cellphone
FK Role
FK Location
Feedback
PK Number
score
comments
FK User
FK Technical
FK Support request
Technical
PK Code
firstname
first lastname
second lastname
email
cellphone
FK Administrator
Notifications
PK Code
message
date
FK User
Location
PK Number
name
number
street
neighborhood
Support request
PK Code
creation date
priority
description
FK User
FK Technical
FK Equipment
FK Tracking history
FK Status
FK Bill
Reports
PK number
description
date
FK Technical
FK Support request
Administrator
PK Code
firstname
first lastname
second lastname
email
cellphone
Status
PK Number
name
Equipment
PK Number
name
description
FK Technical
FK Components
Tracking history
PK Number
code
update date
previuos state
new state
comments
Components
PK Code
name
type