**UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”**

**Group Name:** Jsons

**Members:** Ruben Dario Benavides Macias **NRC:** 9642

Adonny Mateo Calero Argüello

Pablo Esteban Carrera Quinde

Yeshua Amador Chiliquinga Amaya

**System’s requirements**

**1. Introduction**

This document is a Software Requirements Specification (SRS) for the application “OdontoApp”. The content has been created through the collaboration of the members mentioned. This specification has been structured inspired by the guidelines given by the standard "IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998".

**1.1. Purpose**

The objective of carrying out the specification is to clearly and precisely define all the functionalities and restrictions of the system to be built. The document is addressed to the system development team. This document will be reviewed by the developer user group and a record of document versions will be kept on the Git-hub platform, until it is approved by the group involved. Once approved, it will serve as a basis for the development team to build the new system.

**1.2. Scope**

OdontoApp is a platform with the following objective:

Facilitate some dental services to dentists.

At the same time, the system has these following benefits:

- A clinical history that contains necessary medical information that assists the dentist with certain functions.

**1.3. Definitions, Acronyms, and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| FR | It refers to a functional requirement. |
| NFR | It refers to a non functional requirement. |
| GitHub | A Version Control Software (VCS) platform. |
| SRS | Software Requirements Specification |
| PL | Programming Language |
| JAVA | A programming language. |
| csv | Type of files. |

**1.4. References**

| **Document Title** | **Reference** |
| --- | --- |
| Specification of Requirements according to the IEEE 830 standard | https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf |

**1.5. Overview**

Content organization is structured by 4 layers.

In the first one, we have a brief view of the system, the purpose of the present document and some details to know before continuing.

In the second, we can find a broader description of the system and its characteristics, functionalities and a possible future view of the same.

In the third field we explain in a specific level description how the software will work describing technical components and requirements the same has to satisfy.

Finally, in the fourth layer we get extra no related SRS information that complement the document.

**2. Overall Description**

**2.1. Product Perspective**

In general terms, The system will have access to a clinical history that will make available the information of each patient who has gone to the dentist, in order to execute certain actions in benefit of the information management. The system is not related to a larger product, it is completely independent.

**2.2. Product Functions**

The application will manage patient information through a program that allows the dentist to add, delete, edit and list patients with their respective clinical information in an organized manner and with the ability to obtain the said data on file.

**2.3. User Characteristics**

| **User Type** | **Characteristics** |
| --- | --- |
| Dentist | Proper title name  She will be able to create, edit, delete and list a patient and his clinic history. |

**2.4. Constraints**

* The system is the intellectual property of the aforementioned team, which is part of the Universidad de las Fuerzas Armadas ESPE.
* Java PL.
* The development process will involve the use of GitHub and other less-formal communication methods.
* Developers abilities domain: Basic object-oriented programming in Java, files use (csv) in Java, command line management, GitHub.

**2.5. Assumptions and Dependencies**

**2.5.1 Assumptions**

The functionality of the application is expected under the presumption that the requirements mentioned in this document will not be drastically altered since the goals and impact of the application could be directly affected. In the event that changes are required, the OdontoApp development team will agree on any necessary modifications to the system.

**2.5.2 Dependencies**

The application will function autonomously since there is no observed dependence on external applications and/or systems. The system will correspond between the client and the application, and therefore the effectiveness of the system will be directly related to the devices on which the application is used, in this case, only available in a computer

**2.6. Future Requirements**

-For the moment, not defined.

**3. Specific Requirements**

| **Requirement Identifier** | FR01 |
| --- | --- |
| **Scope in the system** | Dentist System Menu |
| **Requirement Definition** | When executing the distributable, the system will first display a menu with the following structure and these available options:  Welcome to OdontoApp, what do you want to do?   1. Add Patient 2. Edit Patient Information 3. Delete Patient 4. Read Patients Information 5. Save in File 6. Exit   Enter an option (with the respective number): |

| **Requirement Identifier** | FR02 |
| --- | --- |
| **Scope in the system** | Dentist System Menu |
| **Requirement Definition** | In the principal menu, the system will be able to enter an option by keyboard explaining limitations (if there are any), otherwise, if the dentist exceeds these ones, the system will send a message explaining the situation and permitting entering an option again. |

| **Requirement Identifier** | FR03 |
| --- | --- |
| **Scope in the system** | Dentist System Menu |
| **Requirement Definition** | After having carried out an action, the system will allow you to return to the menu and the dentist can either carry out another action or exit the application. |

| **Requirement Identifier** | FR04 |
| --- | --- |
| **Scope in the system** | Add Patient Option |
| **Requirement Definition** | Once the dentist has entered the "add patient" option, the system will permit entering the clinical information one by one by keyboard, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR05 |
| --- | --- |
| **Scope in the system** | Edit Patient Information Option |
| **Requirement Definition** | Once the dentist has entered the “edit patient information” option, the system will display another menu with the patients arranged alphabetically and their information, and it will ask for the number of patient (it will be visualizable) to choose to edit his information, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR06 |
| --- | --- |
| **Scope in the system** | Edit Patient Information Option |
| **Requirement Definition** | Once the dentist has chosen the patient, the system will display another menu with the information fields to edit in this way:  Edit “patient name” Information   1. Name 2. Age 3. Weight 4. Height 5. Disease Symptoms 6. Cell Phone Number 7. Systemic Diseases 8. Patient Treatment 9. Treatment Start Date 10. End Date of Treatment   Enter an option: |

| **Requirement Identifier** | FR07 |
| --- | --- |
| **Scope in the system** | Edit Patient Information Option |
| **Requirement Definition** | In the editing patient information menu, when the dentist has entered an option to edit, the system will permit changing the clinical information by keyboard, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR08 |
| --- | --- |
| **Scope in the system** | Edit Patient Information Option |
| **Requirement Definition** | Once the dentist has changed a data, the system will ask if wants to go back to edit another field or if wants to go back to the principal menu with the following structure:  Data has been correctly changed. ¿What do you want to do?   1. Change another data 2. Go back to the principal menu   Enter an option: |

| **Requirement Identifier** | FR09 |
| --- | --- |
| **Scope in the system** | Edit Patient Information Option |
| **Requirement Definition** | Once the dentist is asked to change another field or go back to the principal menu, the system will permit entering an option by keyboard, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR10 |
| --- | --- |
| **Scope in the system** | Delete Patient Option |
| **Requirement Definition** | Once the dentist has entered the “delete patient” option, the system will display another menu with the patients arranged alphabetically and their information, and it will ask for the number of patient (it will be visualizable) to choose to delete, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR11 |
| --- | --- |
| **Scope in the system** | Delete Patient Option |
| **Requirement Definition** | Once the dentist has chosen the patient, the system will ask if wants to go back to delete another patient or if wants to go back to the principal menu with the following structure:  Patient has been correctly deleted. ¿What do you want to do?   1. Delete another patient 2. Go back to the principal menu   Enter an option: |

| **Requirement Identifier** | FR12 |
| --- | --- |
| **Scope in the system** | Delete Patient Option |
| **Requirement Definition** | Once the dentist is asked to delete another patient or go back to the principal menu, the system will permit entering an option by keyboard, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR13 |
| --- | --- |
| **Scope in the system** | Read Patients Information Option |
| **Requirement Definition** | Once the dentist has entered the "read patients information" option, the system will display an arranged alphabetically list of all the patients available and showing a message to go back to the principal menu entering the corresponding number, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR14 |
| --- | --- |
| **Scope in the system** | Save in File Option |
| **Requirement Definition** | Once the dentist has entered the "save in file" option, the system will be able to automatically save the current information of patients in a csv file and showing a message saying if it was saved correctly or not and another one to go back to the principal menu entering the corresponding number, also mentioning if there are limitations when entering data and acting respectively if these have been exceeded. |

| **Requirement Identifier** | FR15 |
| --- | --- |
| **Scope in the system** | Exit Option |
| **Requirement Definition** | Once the dentist has entered the "exit" option, the system will automatically close. |

**3.1. External Interfaces**

**3.1.1. User Interface**

**User interface**

The user interface has a main menu and other additional ones that contain the functions of the system, these being treated with inputs from the dentist's keyboard, these inputs may have limitations (such as passing a data string to a data int), but will be handled properly.

**3.1.2. Hardware Interface**

Previously we will use Yeshua's computer to run and present the program to the Dentist.

**3.1.3. Software Interface**

For the time being, there will be no software interface with other external systems, only in the console or the Windows CMD.

**3.1.4. Communication Interface**

In this case we will save data in .CSV format as this will make it easier to read the data. The end user will be able to enter data and this data will be placed in a cell below the other data.

**3.2. Functions**

**3.2.1** Menu function

**3.2.2** Add patient

**3.2.3** Edit Patient Information

**3.2.4** Delete Patient

**3.2.5** Read Patient Information

**3.2.6** Save the file

**3.2.7** Exit

**3.3. Performance Requirements**

The system will have support for the entry of a person to the application, this will be the client (dentist) and from the application you will be able to manage the data you need about the patients. These data will be saved in a .csv file to provide greater comfort when reading the information. The type of information that is planned to be stored are primitive data types such as string, int, boolean and others. It is expected that this information will be updated as the client uses it, so the information will be open to future changes.

**3.4. Logical Database Requirements**

As mentioned before, we will use csv files; when entering new data each new entry will be placed below the cell used, this means that the user enters a name and the patient's specifications, and this is placed below the previous name entered.

**3.5. Design Constraints**

Our system is limited by Interface constraints, meaning that we will simply restrict ourselves to using the console and displaying the options there. The system will allow data input through a CSV file, and the professional will be able to view it in Excel or in the console. They will also have the ability to modify the desired patients.

**3.6. Software System Quality Attributes**

**3.6.1. Data Validation**

Data validation: ensuring that the data meets the entry requirements by comparing it against a set of pre-established or defined rules. In this case, when entering data in CSV format, we define that each Excel cell will be separated by a ",". This process involves performing a series of checks known as validation routines. It also allows us to better understand the scope of data conflicts.

**4. Appendices**