
Software Requirements Specification

for

CLINIC DENTIX

Version 1.0 approved

Prepared by Group 1

Software Inc

11/03/2024

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References.....	1
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies.....	5
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Hardware Interfaces	5
3.3 Software Interfaces	6
3.4 Communications Interfaces.....	6
4. System Features	6
4.1 System Feature 1.....	6
4.2 System Feature 2 (and so on).....	7
5. Other Nonfunctional Requirements.....	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	7
5.4 Software Quality Attributes	7
5.5 Business Rules	7
6. Other Requirements	8
Appendix A: Glossary.....	8
Appendix B: Analysis Models.....	8
Appendix C: To Be Determined List.....	9

Revision History

Name	Date	Reason For Changes	Version
Group 1	11-03-24	Frist SRS edition	1.0

1. Introduction

1.1 Purpose

This document serves as the Software Requirements Specification (SRS) for the development of a management software tailored for a network of five dental clinics across various EU cities. The software aims to facilitate efficient management of patient records, appointments, treatments, and financial transactions while ensuring compliance with the EU General Data Protection Regulation (GDPR) guidelines.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

The software will be designed to manage patient records, appointments, treatments, and financial transactions across five dental clinics. It will allow interaction for dentists, assistants, and receptionists, each with specific privileges and functionalities. The software will be developed in Java, following the Model-View-Controller (MVC) pattern.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

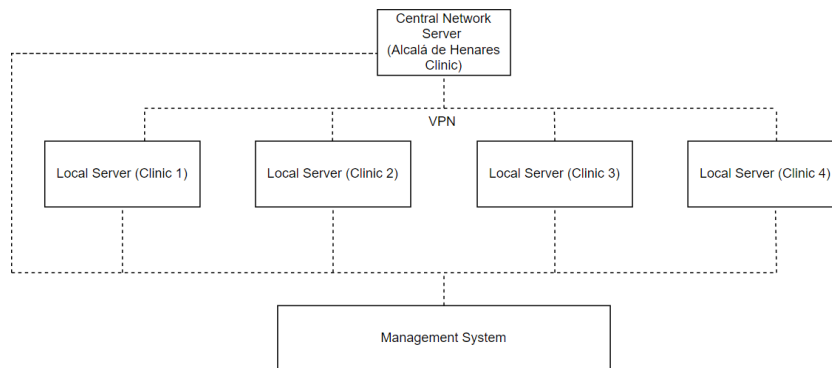
2. Overall Description

2.1 Product Perspective

The product is a new, self-contained management software designed specifically for a network of dental clinics across several EU cities. It is not a follow-on member of a product family nor a replacement for existing systems but rather a novel solution tailored to meet the unique needs and requirements of the dental clinic network.

While this software is self-contained, it operates within the broader context of the dental clinic network, which consists of five clinics interconnected through a central network server located in the clinic in Alcalá de Henares. The software interfaces with local servers at each clinic via LAN connections and communicates with the central network server via VPN using the TCP/IP protocol.

Below is a simplified diagram illustrating the major components of the overall system, including the management software, local servers at each clinic and the central network server.



This diagram illustrates the interconnected nature of the system, with the management software operating at each clinic and interfacing with both local servers and the central network server. The software facilitates communication and data exchange between different components of the network, enabling efficient management of clinic operations.

2.2 Product Functions

Major Functions of the Product:

- Authentication and User Management.
- Patient Record Management.
- Treatment Management.
- Appointment Scheduling.
- Financial Transaction Management.
- Integration with Local and Central Servers.
- Compliance with GDPR Guidelines.
- Language and Localization Support.

2.3 User Classes and Characteristics

User Classes:

1. Dentists:

- **Characteristics:**
 - Highly skilled professionals responsible for diagnosing and treating dental conditions.
 - Require access to patient records for treatment planning and execution.
 - Technically proficient with using software tools for clinical purposes.
- **Privileges:**
 - Full access to patient records for creating, modifying, and deleting.
 - Ability to generate treatment quotes and initiate treatments.

2. Assistants:

- **Characteristics:**
 - Support staff assisting dentists in clinical procedures and patient management.
 - Involved in maintaining patient records and assisting with treatment procedures.
- **Privileges:**
 - Access to patient records for creating and modifying, but not deleting.
 - Assisting dentists in generating treatment quotes and initiating treatments.

3. Receptionists:

- **Characteristics:**
 - Front desk staff responsible for managing appointments, patient inquiries, and administrative tasks.
 - Primary point of contact for patients and visitors.
- **Privileges:**
 - Ability to schedule appointments, modify patient records, print invoices, and collect payments.
 - Limited access to patient records compared to dentists and assistants.

Importance of User Classes:

- Dentists and assistants are the primary users who require extensive access to patient records and treatment functionalities, making them the most important user classes to satisfy.
- Receptionists play a crucial role in managing appointments and administrative tasks, but their access to patient records is more limited compared to dentists and assistants. However, their efficient functioning is still essential for the smooth operation of the clinic.

2.4 Operating Environment

The software will operate within the environment of a networked dental clinic system, encompassing five clinics across several EU cities.

Each clinic has three rooms equipped with modern dental equipment, an x-ray room and a reception room.

Each room has a computer connected to a local server in the reception area via its own LAN and each local server is connected to the central network server located in the clinic in Alcalá de Henares via a VPN, using the TCP/IP protocol, that serves as the hub for interconnecting all clinics and managing data exchange.

2.5 Design and Implementation Constraints

The software must follow the EU General Data Protection Regulation (GDPR) guidelines, imposing strict requirements on the handling and protection of personal data. It must be ensured that all aspects of the software, including data storage, transmission and access controls, adhere to GDPR standards.

The software must have the ability to be integrated within the hardware constraints of the local servers and client computers. The code must meet performance requirements.

Efficient data management strategies must be implemented to minimize memory usage and to optimize performance because the client computers have a limited memory resource that could restrict the amount of data that can be processed and stored.

The software must seamlessly integrate with existing applications and systems within the dental clinics. Standardized interfaces must be implemented to help facilitate data exchange and connectivity between different systems.

The software is required to be developed in Java, respecting the separation of the business model from the user interface (MVC pattern).

To ensure data security, encryption protocols and security measures compliant with industry standards must be implemented.

Role-based access control mechanisms must be implemented to restrict user access to sensitive patient data based on their assigned privileges.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

Third-party components, such as VPN, database management systems, and Java frameworks, are assumed to meet the project's requirements. If the third-party components are not available or do not function as expected, the software architecture may be changed or alternative solutions could be implemented, potentially impacting project timelines and budget.

The development environment, including hardware resources and development tools, is assumed to be sufficient for software development and testing. Limitations in development tools may lead to delays or could affect the quality of the software.

The configuration and performance of local servers and client computers are assumed to support the software's requirements and functionalities. Compatibility issues or performance limitations in the operating environment may affect the software's performance and usability.

The project relies on external factors such as network infrastructure, internet connectivity, and regulatory compliance with GDPR guidelines. Changes in external dependencies may disrupt project activities and require adjustments to be made to the software implementation.

Certain software components or modules may be reused from previous projects or existing libraries to accelerate development. If the assumed software components are not compatible or do not meet the project's requirements, redevelopment may be necessary, leading to delays and increased development effort.

Vendor support for third-party components, including timely updates, bug fixes, and technical assistance, is assumed to be available throughout the project lifecycle. Inadequate vendor support or unexpected changes in vendor policies may hinder the resolution of issues or limit access to necessary resources, potentially impacting project progress and stability.

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

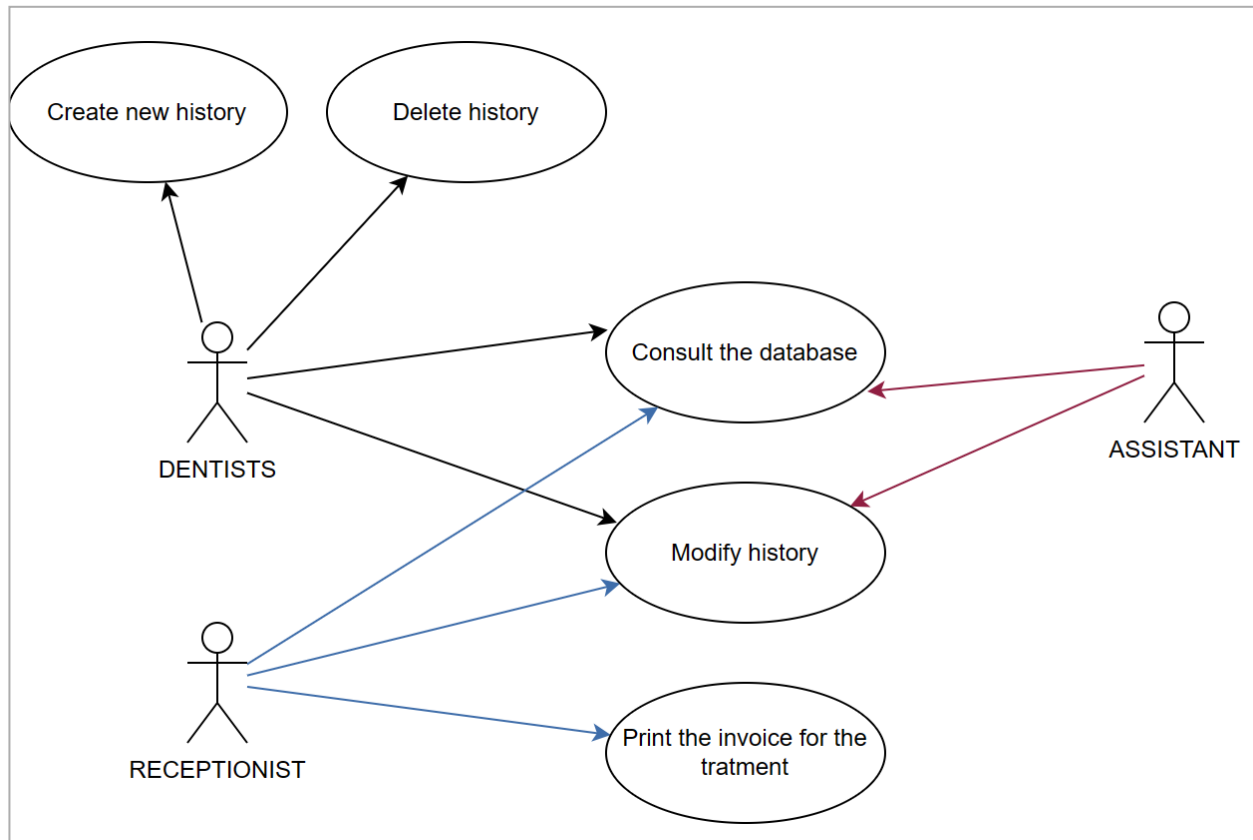
Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

UC-0001	Access of different types of employees to the different functionalities of the system		
Dependencies	<ul style="list-style-type: none">• System access security• Different role employees• Real time updates		
Description	Quick view of the access to the different functionalities of the system and which functionalities could be access as the different types of employees.		
Precondition	The system must implement a filter to know the privileges of the different types of employees and which areas they can access.		
Role and System functionalities	Dentists	<ul style="list-style-type: none">• Can create a new medical history or delate it.• Can modify the medical history at any moment.• Can consult the database.	
	Assistant	<ul style="list-style-type: none">• Can consult the database.• Can modify the medical history.	
	Receptionist	<ul style="list-style-type: none">• Can consult the database.• Can modify the medical history.• Can print the invoice for the treatment.	
Postconditions	The access and changes should be life updated.		

**UC-0001**

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>