Software Requirements Specification

for

CLINIC DENTIX

Version 1.0 approved

Prepared by Group 1

Software Inc

11/03/2024

Table of Contents

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

This document was created using an IEEE 830 standard template written and made available for use by Karl E. Wiegers.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification document is intended for stakeholders involved in the development, implementation, and use of the management system software for the dental clinic network. These stakeholders include developers, project managers, marketing staff, users, and testers.
The document is organized into several sections:
Introduction: Provides an overview of the software project, its objectives, and the scope of the document.
Overall Description: Describes the perspective of the project, mentions the system features, describes the user classes, describes the operating environment, and specifies the constraints, assumptions, and dependencies.
Interfaces: Introduces the interfaces that will be implemented in the software.
System Features: Details and describes the system features with all the necessary requirements.
Nonfunctional Requirements: Details and describes all the nonfunctional requirements.
Appendix: Contains additional information such as the glossary, the use case diagram, the high-class diagram and three use case scenarios with their respective use case and sequence diagrams.

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

This document was written using different examples of SRS documents and various online tutorials for diagram creation.

SRS Example.pdf - IBM (yumpu.com)
SRS.pdf (iastate.edu)
ReqView-Example_Software_Requirements_Specification_SRS_Document.pdf
https://youtu.be/4emxjxonNRI?si=K2f6_oxp3G-vHkG5
https://youtu.be/RHdGn7WMWos?si=HbAoV9Y3y9a1oD37
https://youtu.be/6XrL5jXmTwM?si=6bUzk80i-uk0C0y_
https://youtu.be/pCK6prSq8aw?si=Hd1e0H_30ujL42xR

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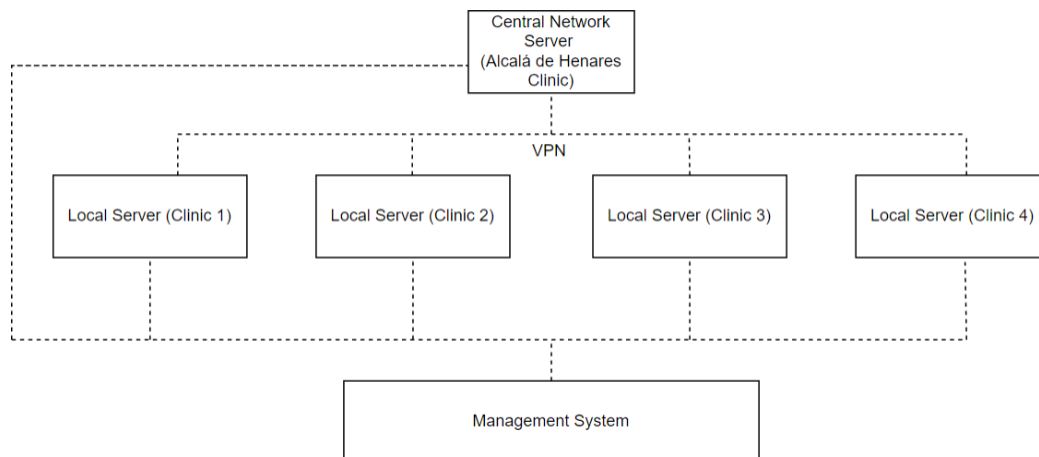
This diagram illustrates the interconnected nature of the system, with the management software operating at each clinic and interfacing with both local servers and the central network server. The software facilitates communication and data exchange between different components of the network, enabling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

2.2 Product Functions

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

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 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

4.1 Authentication and User Management

4.1.1 Description and Priority

Enables user authentication and management of user accounts within the system. High priority.

4.1.2 Stimulus/Response Sequences

User attempts to log in with credentials. System verifies the user credentials and grants access if they are valid.

4.1.3 Functional Requirements

- REQ-1: Provide a log in interface.
- REQ-2: Authenticate users based on their username and password.
- REQ-3: Enforce password complexity requirements.
- REQ-4: Support role-based access control to restrict user access to specific functionalities.
- REQ-5: Allow administrators to create, modify and delete user accounts.

4.2 Management of Patient History

4.2.1 Description and Priority

Allows the management of patient records, including creation, modification and deletion. High priority.

4.2.2 Stimulus/Response Sequences

User selects a patient record for viewing or editing. If the record exists, the system displays the patient's record and allows for modifications or deletion. If the record does not exist, the system allows creation.

4.2.3 Functional Requirements

- REQ-6: Provide a search function to locate patient records based on patient information.
- REQ-7: Allow users to create or delete new patient records.
- REQ-8: Support all modifications made to the patient history.
- REQ-9: Restrict deletion or creation of patient records to authorized personnel only.
- REQ-10: Ensure confidentiality and integrity of patient data in accordance with EU GDPR guidelines.

4.3 Treatment Management

4.3.1 Description and Priority

Management of treatments provided to patients, including recording treatment details and status. Medium priority.

4.3.2 Stimulus/Response Sequences

User selects a patient's record to view treatment history. System displays the treatment history and allows for updates to be made.

4.3.3 Functional Requirements

- REQ-11: Allow users to record details of treatments provided to patients.
- REQ-12: Support the tracking of treatment status.
- REQ-13: Allow for association of treatments with patient records.

4.4 Appointment Scheduling

4.4.1 Description and Priority

Allows receptionists to schedule appointments for patients. High priority.

4.4.2 Stimulus/Response Sequences

Receptionist selects the appointment date and time. System confirms the appointment and updates the schedule.

4.4.3 Functional Requirements

- REQ-14: Display a calendar interface for selecting appointment dates.
- REQ-15: Allow receptionist to input patient information.
- REQ-16: Validate appointment times to prevent double bookings.
- REQ-17: Send appointment confirmation to patients.
- REQ-18: Allow the receptionist to update the appointment details if the appointment needs to be rescheduled.

4.5 Financial Transactions Management

4.5.1 Description and Priority

Management of financial transactions, including invoicing and payment processing. High priority.

4.5.2 Stimulus/Response Sequences

Receptionist selects a completed treatment for invoicing. System generates and displays the invoice for the selected treatment.

4.5.3 Functional Requirements

- REQ-19: Calculate the total cost of a treatment based on treatment details and the quotes provided by the dentist.
- REQ-20: Generate itemized invoices listing the treatment costs and the patient information.
- REQ-21: Allow receptionists to print invoices for patients.
- REQ-22: Update the treatment status to "paid" when the payment is received.
- REQ-23: Support multiple payment methods (cash/card).
- REQ-24: Provide a receipt for the payments received and update the patient's financial record accordingly.

4.6 Compliance with GDPR Guidelines

4.6.1 Description and Priority

Ensure that the system complies with the EU General Data Protection Regulation guidelines for handling personal data. High priority.

4.6.2 Stimulus/Response Sequences

User attempts to access or modify personal data. System validates user permissions and enforces GDPR compliance measures.

4.6.3 Functional Requirements

- REQ-25: Encrypt all personal data stored in the database to protect confidentiality.
- REQ-26: Implement access controls to restrict unauthorized access to personal data.
- REQ-27: Provide mechanisms for users to review and update their consent for data processing.
- REQ-28: Log all access and modifications to personal data for reviewing purposes.
- REQ-29: Allow patients to request access to their personal data and provide mechanisms for data portability.
- REQ-30: Have procedures in place for timely reporting of data breaches to regulatory authorities and affected individuals.
- REQ-31: Regularly conduct data protection impact assessments to identify and mitigate risks.

4.7 Integration with Local and Central Servers

4.7.1 Description and Priority

The integration of the system with both local servers within each clinic and a central network server located in Alcalá de Henares. High priority.

4.7.2 Stimulus/Response Sequences

System initiates a data transfer request to the local server. Local server processes the request and sends back the requested data.

4.7.3 Functional Requirements

- REQ-32: Establish and maintain a secure connection to the local server in each clinic via LAN.
- REQ-33: Enable consultation of patient records stored on the local server.
- REQ-34: Establish a VPN connection to the central network server.
- REQ-35: Use the TCP/IP protocol for communication between local and central servers.
- REQ-36: Synchronize patient data between local and central servers to ensure consistency across all clinics.
- REQ-37: Allow for real time data access and updates from any clinic location.
- REQ-38: Have mechanisms in place to ensure data availability in case of server failures.
- REQ-39: Prioritize data transmission based on network bandwidth availability to optimize performance.

4.8 Language Localization Support

4.8.1 Description and Priority

Support multiple languages for users across the different countries of the clinics. Medium priority.

4.8.2 Stimulus/Response Sequences

User selects a preferred language from the language options. System displays the user interface and content in the selected language.

4.8.3 Functional Requirements

- REQ-40: Provide language selection options in the user interface.
- REQ-41: Support localization of user interface elements into multiple languages.
- REQ-42: Allow administrators to configure default language settings for each clinic.
- REQ-43: Support dynamic language switching without requiring system restarts or user logouts.
- REQ-44: Ensure consistency in language translations across all modules and screens.
- REQ-45: Provide language specific formatting options for dates, times and currency.

4.9 Taxes Handling

4.9.1 Description and Priority

Accommodate varying tax rates and regulations in the countries where the dental clinics are located. Medium priority.

4.9.2 Stimulus/Response Sequences

System identifies the country where the clinic is located during transaction processing. System applies the appropriate tax rates and calculations based on the country's tax regulations.

4.9.3 Functional Requirements

- REQ-46: Support configuration of tax rates for each clinic's country.
- REQ-47: Allow administrators to update tax rates and regulations as per changes in local tax laws.
- REQ-48: Apply the relevant tax rates to financial transactions based on the clinic's location.
- REQ-49: The applied taxes should be reflected in the generated invoice for compliance and accounting purposes.
- REQ-50: Provide alerts or notifications to users when tax rates or regulations are updated or changed.
- REQ-51: Maintain a tax history log to track tax rate changes and adjustments over time.
- REQ-52: Ensure that tax calculations are accurate and comply with local tax authoroties' requirements.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response time – Respond to user interactions within 2 seconds on average.

Throughput – Be capable of handling concurrent access from at least 20 users per clinic without a noticeable decrease in performance.

Scalability – Accommodate additional clinics and users without significant degradation in performance. It should handle at least a 20% increase in clinics and users without requiring hardware updates.

Reliability – Have an uptime of at least 99.9%, ensuring the system is available for use during clinic hours without interruptions.

Data Integrity – All data transactions should be securely stored and maintained to prevent data loss or corruption. Data backups should be performed nightly and data recovery procedures should be tested regularly.

5.2 Safety Requirements

Backup and Recovery – Regular backups of the system must be performed to prevent data loss in case of system failure or cyber-attacks. A data recovery plan should be implemented to ensure continuity of operation in case data loss happens.

Error Handling – Resilient to errors and failures, robust error handling mechanisms should be implemented to prevent data corruption or system instability.

Training and Documentation – Adequate training and documentation should be provided to users to ensure they understand how to use the system safely and effectively.

5.3 Security Requirements

User Authentication – Users must authenticate themselves before accessing the system. Authentication should be based on a secure username/password authentication mechanism.

Secure Communication – Communication between the local servers and the central network server must be encrypted using TLS to prevent eavesdropping or tampering with data transmitted over the network. Secure VPN connections should be established to ensure data privacy and integrity.

Security Certification – The software should undergo security assessments to validate its compliance with industry standards. Certifications such as ISO 27001 or SOC 2 may be required to demonstrate the software's adherence to security protocols.

Data Confidentiality – All patient data should be encrypted both in transit and at rest to prevent unauthorized access or data breaches.

Access Control – Different levels of access should be defined for dentists, assistants, and receptionists, with access rights limited to the necessary functions for each role.

Auditability and Accountability – Maintain detailed logs of user actions. These logs should be tamper-proof and accessible for auditing purposes to ensure accountability and traceability of all activities performed within the system.

5.4 Software Quality Attributes

Accessibility – Ensure that the user interface is accessible to individuals with disabilities, conforming to relevant accessibility standards such as WCAG (Web Content Accessibility Guidelines). Provide alternative input methods and assistive technologies to support users with diverse needs.

Availability – The software should be available to use during clinic operating hours (8:00 AM to 6:00 PM). Downtime for maintenance should be scheduled during non-working hours.

Reliability – The mean time between failures should be of at least 500 hours. Failures should be rare, and the system should be capable of recovering promptly from unexpected errors or crashes.

Maintainability – The software should allow for easy updates, bug fixes and enhancements. Code should be well documented, modular, and organized to facilitate future maintenance by developers.

Usability – The software should be intuitive and user-friendly, new users should be able to learn the basic functionality of the system in 30 minutes. User interfaces should be designed with clear navigation, informative feedback, and a consistent layout.

Interoperability – The software should be interoperable with other systems and applications commonly used in dental clinics (for example dental imaging software or electronic health record systems). It should support standard data exchange formats and protocols to facilitate seamless integration with external systems.

Portability – The software should be portable across different operating systems and hardware platforms commonly used in dental clinics. It should be designed using platform-independent technologies to ensure compatibility and ease of deployment.

Robustness – The software should be resilient to errors or invalid inputs, with error-handling mechanisms in place to prevent system crashes or data corruption.

Testability – The software should allow for comprehensive testing of all system functionalities and components. Test cases should be automated wherever possible to streamline the testing process and ensure thorough coverage.

5.5 Business Rules

Patient History and Appointments:
- Dentists can create, modify, and delete patient records.
- Assistants can create and modify patient records.

- Receptionists can modify patient records and make appointments.

Treatment and Payment:
- Dentists can generate treatment quotes and initiate treatments.
- Receptionists can print treatment invoices and collect payments.

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	• 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	• Can create a new medical history or delate it. • Can modify the medical history at any moment. • Can consult the database.
	Assistant	• Can consult the database. • Can modify the medical history.
	Receptionist	• 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.	
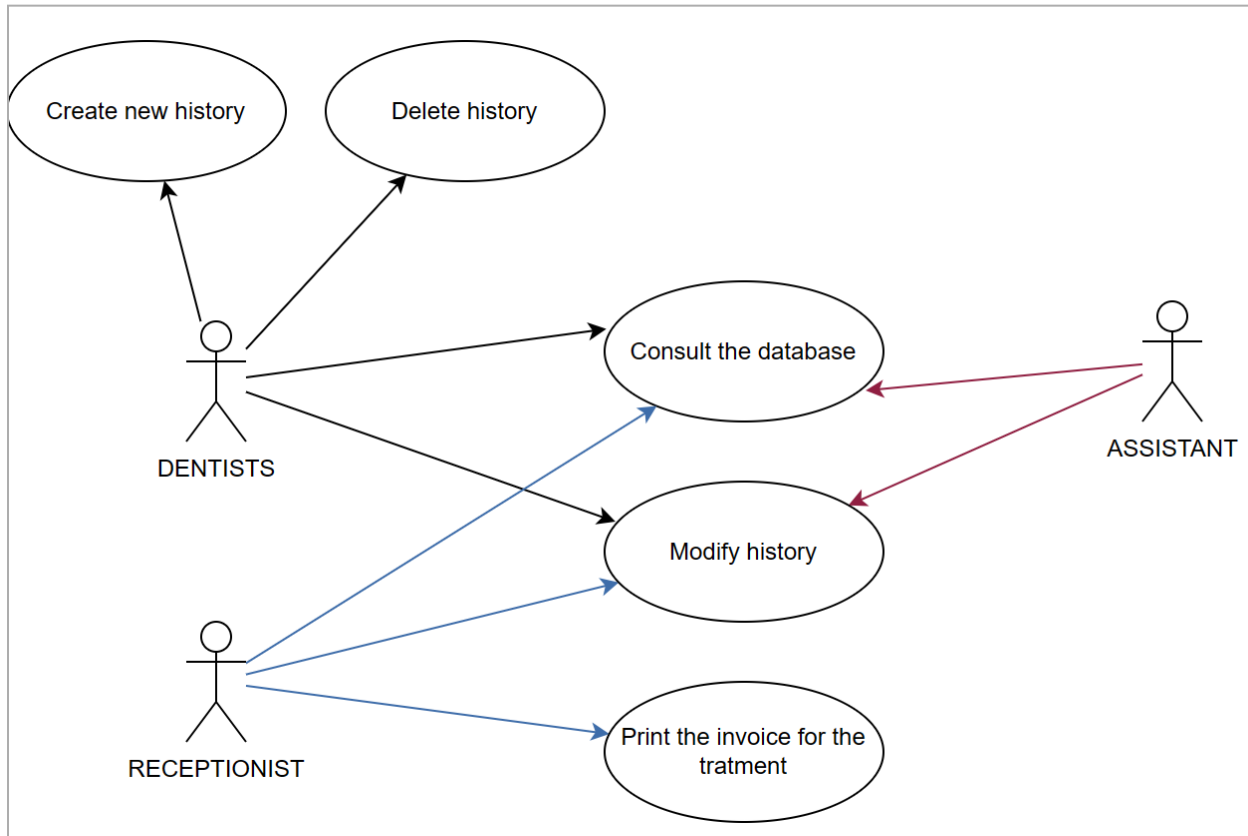
UC-0001

UC-0003	Access of different types of employees to the different functionalities of the system
Dependencies	• System access security • Different role employees • Real time updates • Maintenance
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 log-in system that discriminates access to certain features based on user type.
	Dentists • Can manage all aspects of the patient history system.

Role and System functionalities		• Can generate quotes. • Can log-in
	Assistant	• Can view, modify and create patient history. • Can modify the medical history.
	Receptionist	• Can access the invoice system. • Can modify the patient history. • Can access the appointment system
Postconditions	The access and changes should be life updated.	

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.>