# Dental Surgery Appointment Management System (DAMS) - Requirements Specification

## 1. Functional Requirements (FR)

The system's core functionality is categorized below, covering patient, provider, and appointment management, highlighted by the advanced AI-powered chatbot interface.

---

### 1.1 Patient Management

| ID | Requirement Description |
|---|---|
| **FR1.1** | The system shall allow the **creation of new patient records** with a unique patient identifier (patient number). |
| **FR1.2** | The system shall store comprehensive patient information, including name, patient number, contact details, and address. |
| **FR1.3** | The system shall allow **searching for patients** by name or patient number. |
| **FR1.4** | The system shall display the complete **appointment history** for any selected patient. |
| **FR1.5** | The system shall allow the **updating of patient information**. |
| **FR1.6** | The system shall allow the **deletion of patient records**. |

### 1.2 Dentist Management

| ID | Requirement Description |
|---|---|

| | |
|---|---|
| **FR2.1** | The system shall allow the **creation of dentist profiles** including their specialization. |
| **FR2.2** | The system shall store dentist information, including name, specialization, contact phone, and address. |
| **FR2.3** | The system shall allow **searching for dentists** by name or specialization. |
| **FR2.4** | The system shall **associate dentists with one or more surgery locations**. |
| **FR2.5** | The system shall display the complete **appointment schedule** for any selected dentist. |
| **FR2.6** | The system shall allow the **updating of dentist information**. |

## 1.3 Appointment Management

| **ID** | **Requirement Description** |
|---|---|
| **FR3.1** | The system shall allow the creation of appointments, specifying the patient, dentist, date/time, and surgery location. |
| **FR3.2** | The system shall **validate appointment conflicts** (e.g., double-booking, scheduling outside hours) before final creation. |
| **FR3.3** | The system shall display a comprehensive list of **all appointments** in the system. |
| **FR3.4** | The system shall allow **filtering appointments** by patient, dentist, or date. |

| | |
|---|---|
| **FR3.5** | The system shall allow the **updating of appointment details** (e.g., rescheduling). |
| **FR3.6** | The system shall allow the **cancellation of appointments**. |

## 1.4 AI-Powered Chatbot

| ID | Requirement Description |
|---|---|
| **FR4.1** | The system shall provide a **conversational chatbot interface** for users. |
| **FR4.2** | The chatbot shall understand **natural language queries** using Google Gemini AI. |
| **FR4.3** | The chatbot shall **extract core entities** (patient name, dentist name, appointment time) from user messages. |
| **FR4.4** | The chatbot shall execute **patient searches** through conversational queries (e.g., "Find patient John Doe"). |
| **FR4.5** | The chatbot shall execute **dentist searches** through conversational queries (e.g., "Find dentist by specialty"). |
| **FR4.6** | The chatbot shall **create appointments** through natural language commands (e.g., "Book appointment for..."). |
| **FR4.7** | The chatbot shall list all patients, dentists, or appointments upon request. |
| **FR4.8** | The chatbot shall provide **help and guidance** on available commands and supported queries. |

| | |
|---|---|
| **FR4.9** | The system shall **fall back to rule-based processing** immediately if the AI service fails or is unavailable. |
| **FR4.10** | The chatbot shall provide **quick action buttons** for common tasks (e.g., "Make Appointment," "Search Dentist"). |

## 1.5 Surgery Management

| ID | Requirement Description |
|---|---|
| **FR5.1** | The system shall maintain **surgery location records** with unique surgery identifiers. |
| **FR5.2** | The system shall associate each surgery with a complete street address. |
| **FR5.3** | The system shall track and display the list of **dentists assigned to each surgery location**. |
| **FR5.4** | The system shall display appointments scheduled specifically at each surgery. |

## 1.6 Address Management

| ID | Requirement Description |
|---|---|
| **FR6.1** | The system shall store complete, structured **address information** (street, city, state, zip code). |
| **FR6.2** | The system shall associate addresses with **patients, dentists, and surgeries**. |

|  |  |
|---|---|
| **FR6.3** | The system shall allow **updating of address information** associated with any record. |

## 2. Non-Functional Requirements (NFR)

These requirements define the quality attributes and performance constraints of the system.

### 2.1 Performance

| ID | Requirement Description |
|---|---|
| **NFR1.1** | The **chatbot shall respond to user queries within 3 seconds** under normal operating load. |
| **NFR1.2** | The AI processing shall complete within **5 seconds**; otherwise, the system must trigger the rule-based fallback. |
| **NFR1.3** | The system shall support **concurrent access** and full functionality for multiple simultaneous users (receptionists). |

### 2.2 Usability

| ID | Requirement Description |
|---|---|
| **NFR2.1** | The chatbot interface shall be **intuitive, conversational**, and minimize the learning curve for staff. |
| **NFR2.2** | The system shall provide **clear, constructive error messages** and suggestions for user actions. |
| **NFR2.3** | The user interface (UI) shall be **responsive**, fully functional, and well-designed on both desktop and mobile browsers. |

### 2.3 Reliability

| ID | Requirement Description |
|---|---|
| **NFR3.1** | The system shall maintain **99% uptime availability** during business hours. |
| **NFR3.2** | The chatbot shall **gracefully handle AI service failures** by automatically switching to rule-based fallback processing (see FR4.9). |
| **NFR3.3** | The system shall prevent data loss through proper database transaction management. |

## 2.4 Security

| ID | Requirement Description |
|---|---|
| **NFR4.1** | The system shall **authenticate users** (login/password) before allowing any access to features. |
| **NFR4.2** | The system shall **protect sensitive patient and dentist information** using appropriate security measures (e.g., encryption, access control). |
| **NFR4.3** | API keys for external services (e.g., Gemini AI) shall be securely stored and managed. |

## 2.5 Maintainability

| ID | Requirement Description |
|---|---|
| **NFR5.1** | The code base shall adhere to established **Spring Boot and React best practices and coding standards**. |

| | |
|---|---|
| **NFR5.2** | The system shall use **Data Transfer Objects (DTOs)** for API communication to prevent circular reference issues and structure payloads effectively. |
| **NFR5.3** | The system shall implement robust **logging** for all errors and important system events for effective debugging and auditing. |

# 3. Use Cases

The following use cases detail the interaction between the actors and the system for critical features, focusing heavily on the AI-enabled workflow.

### Use Case 3.1: Book Appointment via Chatbot

| **Detail** | **Description** |
|---|---|
| **Actor** | Receptionist / User |
| **Precondition** | Patient and dentist records exist in the system. |
| **Main Flow** | 1. User types: "Book appointment for **Gillian White** with **Tony Smith tomorrow at 2pm**." 2. System uses AI to extract Patient, Dentist, and Date/Time entities. 3. System searches/validates patient and dentist existence. 4. System validates the proposed date/time and checks for conflicts (FR3.2). 5. System creates and saves the new appointment record (FR3.1). 6. System displays a success confirmation card with appointment details. |
| **Alternative Flows** | **3a. Entity Not Found:** System prompts the user to search for the patient or dentist first. **4a. Conflict:** System displays an error and suggests alternative available times. **4b. Invalid Format:** System requests clarification on the date/time format. |

| | |
|---|---|
| **Postcondition** | A new, validated appointment record is created and saved in the system. |

### Use Case 3.2: Search Patient via Chatbot

| **Detail** | **Description** |
|---|---|
| **Actor** | Receptionist / User |
| **Precondition** | User is logged into the system. |
| **Main Flow** | 1. User types in chatbot: "Find patient **Gillian White**." 2. System processes message with AI (or rule-based parsing). 3. System searches the patient database for the extracted name. 4. System displays patient information (including address and contact) (FR1.3). 5. System provides quick action buttons (e.g., "Make appointment," "View History"). |
| **Alternative Flows** | **2a. AI Unavailable:** System automatically uses rule-based keyword matching (FR4.9). **4a. No Match:** System suggests listing all patients or re-entering the name. |
| **Postcondition** | User successfully views patient search results. |

### Use Case 3.3: AI Service Fallback

| **Detail** | **Description** |
|---|---|
| **Actor** | System |
| **Precondition** | User sends any message to the chatbot. |

| | |
|---|---|
| **Main Flow** | 1. System attempts to process the message via the Gemini AI service. 2. **AI service call fails** or exceeds the 5-second timeout (NFR1.2, NFR3.2). 3. System logs the AI failure event. 4. System automatically and instantly **falls back to rule-based processing** (FR4.9). 5. System processes the user message using keyword matching for common commands (e.g., "list patients," "help"). 6. System returns the appropriate rule-based response. |
| **Postcondition** | User receives a response despite the AI service failure, ensuring continuity of service. |

### Use Case 3.4: Create Appointment (Manual Form)

| **Detail** | **Description** |
|---|---|
| **Actor** | Receptionist |
| **Precondition** | User is logged in, patient and dentist records exist. |
| **Main Flow** | 1. User navigates to the dedicated "Create Appointment" page. 2. User selects **Patient, Dentist, and Surgery Location** from dropdowns. 3. User selects **Date and Time** via calendar and time pickers. 4. User submits the form. 5. System validates input and checks for scheduling conflicts (FR3.2). 6. System creates and saves the appointment record (FR3.1). 7. System displays a success confirmation message. |
| **Alternative Flows** | **5a. Validation Fail:** System highlights all invalid fields and shows an error message. **5b. Conflict:** System displays an error and prompts the user to select an alternative slot. |
| **Postcondition** | A new appointment is created and saved via the manual staff interface. |