Toothbook.ca Software Requirements Specification

Authors:

* Arifali Baghirli - 210201848
* Zoher Matar - 210201919
* Yahye Said Abdi - 210201922
* Bashir Tayani - 220201812
* Abdulbari Abdulghani Ahmed Leftesi - 220201877
* Ramiz Shafiyev - 220201988
* Bashar Ali Mohamed Osman - 220201863
* Ahmed Mohamed Ahmed – 200201874

Table of Contents:

**1. Introduction**

* 1.1 Purpose
* 1.2 Scope
* 1.3 Definitions, Acronyms, and Abbreviations
* 1.4 References
* 1.5 Overview (of the document)

**2. Overall Description**

* 2.1 Product Perspective
* 2.2 Product Functions
* 2.3 User Classes and Characteristics
* 2.4 Operating Environment
* 2.5 Design and Implementation Constraints
* 2.6 User Documentation
* 2.7 Assumptions and Dependencies

**3. Specific Requirements**

* 3.1 Functional Requirements (detailed by modules or use cases)
* 3.2 External Interface Requirements
  + User Interfaces
  + Hardware Interfaces
  + Software Interfaces
  + Communication Interfaces
* 3.3 Non-functional Requirements
  + Performance
  + Security
  + Reliability
  + Usability
  + Maintainability
  + Portability

**1.1 Purpose**

The purpose of this document is to define the software requirements for the **Toothbook Online Clinic Appointment System**. This system will allow patients to search and find nearest dental clinics in Canada, view available appointment slots, and book or manage their appointments online. It also provides functionality for clinics and administrators to manage schedules, patient records, and feedback.

This document is intended for software developers, testers, project stakeholders, and system designers involved in the development and maintenance of the system.

**1.2 Scope**

Toothbook is a web-based clinic appointment system designed to streamline the booking and management of dental appointments. The system supports the following main functionalities:

* Patient registration and login
* Search and filter clinics by dental services, location, specialty, or rating
* Book, cancel, and reschedule appointments
* View dentist/clinic profiles and reviews
* Leave feedback after the appointment
* Chat with clinic, patient and administration
* Admin and clinic dashboards for schedule and user management
* Notification system (Email, Telegram) SMS is planned for future
* Automated Background Tasks for Marketing
* Charge clinic cards automatically/manually over Stripe

The system will be accessible via web browsers (desktop/mobile) and developed using modern web technologies. It aims to improve patient satisfaction, clinic efficiency, and reduce no-shows through reminders and scheduling features.

**1.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| UI | User Interface |
| UX | User Experience |
| API | Application Programming Interface |
| DBMS | Database Management System |
| JWT | JSON Web Token (used for authentication) |
| OTP | One-Time Password |
| SRS | Software Requirements Specification |
| CRUD | Create, Read, Update, Delete |

**1.4 References**

* IEEE 830-1998: Recommended Practice for Software Requirements Specifications
* Microsoft SQL Server official documentation ([Microsoft SQL Server](https://learn.microsoft.com/sr-latn-rs/sql/?view=sql-server-ver16))
* ASP.NET Core documentation ([ASP .NET Core 8](https://learn.microsoft.com/en-us/aspnet/core))
* React.js documentation ([React JS](https://reactjs.org/))
* TypeScript Documentation ([TypeScript](https://www.typescriptlang.org/docs/))
* Plesk Obsidian Documentation ([Plesk](https://docs.plesk.com/en-US/obsidian/))
* IIS Server Documentation ([IIS](https://learn.microsoft.com/en-us/iis/))
* Node JS Documentation ([NodeJS](https://nodejs.org/docs/latest/api/))
* ISO 27001:2022 ([ISO27001](https://www.nqa.com/getmedia/ae12c945-4dbb-4b73-a4e3-996261a540af/NQA-ISO-27001-Implementation-Guide.pdf))
* General Data Protection Regulation ([GDPR](https://gdpr-info.eu/))
* Figma Documentation ([Figma](https://www.figma.com/community/file/944644388375583520/ux-documentation))

**1.5 Overview of Document**

This document outlines the functional and non-functional requirements of the system. It includes:

* An overall description of the system and its environment
* Detailed functional and non-functional requirements
* User and system interactions
* External interface requirements

**2. Overall Description**

This section gives a high-level understanding of the system’s context, users, constraints, and environment before diving into detailed requirements. Here's the standard structure:

**2.1 Product Perspective**

* Is this a new system or part of a larger system?
* Describe the product's position in the ecosystem.
* Reference existing systems it interacts with.

**2.2 Product Functions**

Summarize the major functions of the system:

* Patient registration and login
* Booking and managing appointments
* Clinic search with filters
* Review and rating system
* Admin dashboard for clinic/user management
* Email/SMS notifications
* Charge clinic cards automatically over Stripe
* Chat with patients, clinics, admin users
* Send Marketing emails
* Authorize and authenticate users over OAuth2

**2.3 User Classes and Characteristics**

List and describe the different user types:

* **Patient**: books appointments, writes reviews, receives notifications, chat with clinic
* **Dentist, Clinic Manager Clinic Admin**: manages schedule, sees reviews, handles patient data, chat with patients, system administrators
* **System Admin**: full control, manages clinics and users  
  Include special access needs or technical experience if any.

**2.4 Operating Environment**

* Web browser (Chrome, Firefox, Safari)
* Server OS (Windows Server, Shared Hosting)
* Backend: .NET Core Web API
* Frontend: React.js TypeScript
* Database: Microsoft SQL Server
* Hosting: Shared Hosting (migration planned to Windows VPS)

**2.5 Design and Implementation Constraints**

* Must use HTTPS, HSTS, SSL/TLS for all communications
* JWT for authentication
* Mobile-responsive UI
* Must support at least 1000 concurrent users
* Compliance with local data privacy laws

**2.6 User Documentation**

* Online user guide (for patients and clinics)
* Help section in dashboard
* About, FAQ and support contact page

**2.7 Assumptions and Dependencies**

* Users have internet access and an email address
* Clinics are responsible for updating their schedules
* External Email service providers are available and stable

**3. Specific Requirements**

This section defines all the **functional**, **non-functional**, and **interface** requirements in detail.

**3.1 Functional Requirements**

Each functional requirement can be written as a **use case** or structured list.

**Patient Functions**

* **FR1**: The system shall allow patients to register with email, password, name, and phone number.
* **FR2**: The system shall allow patients to log in securely using JWT-based authentication.
* **FR3**: The system shall allow patients to search clinics by name, location, specialty, and rating.
* **FR4**: The system shall allow patients to view clinic profiles, available time slots, and reviews.
* **FR5**: The system shall allow patients to book appointments from available time slots.
* **FR6**: The system shall allow patients to cancel or reschedule appointments.
* **FR7**: The system shall allow patients to post reviews and ratings after attending appointments.
* **FR8**: The system shall send email/SMS reminders before the appointment.

**Clinic/Dentist Functions**

* **FR9**: Clinics shall be able to create and manage available time slots.
* **FR10**: Clinics shall be able to view upcoming appointments.
* **FR11**: Clinics shall be able to manage their profile information and services.
* **FR12**: Clinics shall receive appointment notifications.

**Admin Functions**

* **FR13**: Admin shall manage all registered clinics and users.
* **FR14**: Admin shall deactivate or approve clinic accounts.
* **FR15**: Admin shall view system analytics (appointments, usage).

**3.2 External Interface Requirements**

**User Interface**

* Responsive web design (desktop/tablet/mobile)
* Simple, modern UI for patients
* Dashboard-style UI for clinics and admins

**API Interfaces**

* RESTful APIs (secured with JWT)
* Example:
  + POST /api/Identity/Account/Login
  + PATCH /api/Identity/Account/UpdateAccount
  + GET /api/BookingOverview/GetBooking
  + GET /api/Clinic/SearchClinics

**Software Interfaces**

* Database: Microsoft SQL Server (entity relationships and stored data)
* Email/SMS API: (e.g., SMTP, Telegram)
* Google Maps API (for location-based search)
* Visual Studio
* Visual Studio Code

**Communication Interfaces**

* HTTPS for all client-server interactions
* REST over HTTP/1.1

**3.3 Non-Functional Requirements**

**Performance**

* System shall respond to user actions within 2 seconds for 90% of operations.
* Shall support at least 500 concurrent users.

**Security**

* All data transfers shall be encrypted using HTTPS.
* User passwords shall be hashed and stored securely.
* Only authenticated users can access protected endpoints.
* JWT Tokens shall be revoked once the suspicious activity reported.

**Usability**

* Interface shall be user-friendly with consistent design across pages.

**Reliability**

* The system shall have at least 99.5% uptime per month.

**Maintainability**

* System shall be modular and follow layered architecture.

**Scalability**

* Designed to support future mobile app integration and cloud scaling.