Software Requirements Specification

for

<Beauty Salon Appointment System>

Version 1.0 approved

Prepared by <Ebrar İkbal Çalışkan>

<organization>

<date created>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to define all functional and non-functional requirements for the *Beauty Salon Appointment System*.
The system aims to digitalize the manual appointment management process of beauty salons to increase customer satisfaction and staff efficiency.

1.2 Document Conventions

All requirements are numbered as "REQ-x". Each requirement is classified as **High**, **Medium**, or **Low** priority.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, testers, project managers, UI designers, and system administrators.
Readers should begin with the general overview (Section 2) and continue with detailed system features (Section 4).

1.4 Product Scope

The Beauty Salon Appointment System enables customers to book, cancel, or modify appointments online.
Employees can view and approve their assigned appointments, while administrators manage services and staff.
The system also sends automated reminder emails to customers before their appointment time.

1.5 References

- Ebrar İkbal Çalışkan, *Software Engineering UML Homework (2025)*
- IEEE Std 830-1998, *Recommended Practice for Software Requirements Specifications*

2. Overall Description

2.1 Product Perspective

The system is an independent web/mobile application.
All data are stored in a cloud-based relational database (e.g., MySQL or PostgreSQL).

2.2 Product Functions

- *Book, cancel, and view appointments*
- *Display available services and prices*
- *Allow employees to confirm or reject appointments*
- *Enable administrators to manage services and employees*
- *Send automatic email or push reminders*

2.3 User Classes and Characteristics

User Type	Description
Customer	Selects services and books appointments.
Employee	Views and approves/rejects assigned appointments.
Administrator	Manages employees, services, and system data.
System	Sends automated reminders and notifications.

2.4 Operating Environment

- **Web:** HTML5, CSS, JavaScript
- **Mobile:** Flutter or React Native
- **Backend:** Node.js / Django REST
- **Database:** MySQL / PostgreSQL

2.5 Design and Implementation Constraints

- User authentication is mandatory.
- Appointment time conflicts must be prevented.
- Time zones must be synchronized across devices.

2.6 User Documentation

- User Manual (PDF)
- FAQ section
- Optional tutorial videos

2.7 Assumptions and Dependencies

- Users have an active internet connection.
- Email reminders use an SMTP service.
- Staff working hours are predefined in the system.

3. External Interface Requirements

3.1 User Interfaces

- Homepage displaying services and prices
- “Book Appointment” form (service, employee, date, time)
- Admin panel for CRUD operations
- Email reminder templates

3.2 Hardware Interfaces

- Compatible with desktop and mobile devices
- Minimum 2 GB RAM, 1.5 GHz CPU

3.3 Software Interfaces

- Database connection (SQL)
- SMTP mail service
- Optional: Google Calendar API integration

3.4 Communications Interfaces

- HTTPS REST API for client–server communication
- Optional WebSocket for live appointment updates

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 Book Appointment

Priority: High
Description: Customers select a service and book a time slot.
Stimulus/Response Sequence:

1. ***Customer selects a service.***
2. ***The system lists available employees and time slots.***
3. ***Customer confirms the booking.***
4. ***System stores the appointment and sends a confirmation email.***

Functional Requirements:

REQ-1: The system shall verify time slot availability.

REQ-2: A confirmation message shall be displayed upon successful booking.

4.2 Cancel Appointment

Priority: Medium
Description: Customers can cancel their existing appointments.

REQ-3: The system may request a cancellation reason.

REQ-4: Both the assigned employee and admin shall be notified after cancellation.

4.3 Manage Services

Priority: High

REQ-5: Administrators shall be able to add, update, or delete services.

4.4 Approve Appointment

Priority: High

REQ-6: Employees shall approve or reject assigned appointments.

REQ-7: If rejected, the system shall suggest alternative time slots to the customer.

4.5 Send Reminder (System Automation)

Priority: High

REQ-8: The system shall automatically send an email reminder 24 hours before the appointment.

REQ-9: An optional SMS reminder feature may be integrated (alternative requirement).

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Each operation shall respond within 2 seconds.

The system shall support at least 100 concurrent users.

5.2 Safety Requirements

Appointment data must not be lost.

Personal data must remain secure after logout.

5.3 Security Requirements

All user actions require authentication.

Passwords shall be hashed using SHA-256.

Unauthorized access shall trigger a generic “Access Denied” message.

5.4 Software Quality Attributes

Usability: Simple and intuitive interface

Maintainability: Modular architecture

Reliability: 99% uptime target

Scalability: Easily extendable for new services

5.5 Business Rules

Each customer can have only one appointment per time slot.

Employees can approve only their own assigned appointments.

6. Other Requirements

- The system should provide multilingual support (alternative requirement).
- A customer rating and feedback system may be added (UX enhancement).
- A deposit or pre-payment option may be offered for future appointments.

Appendix A: Glossary

- ***CRUD:*** *Create, Read, Update, Delete*
- ***UI:*** *User Interface*
- ***SMTP:*** *Simple Mail Transfer Protocol*

Appendix B: Analysis Models

- *UML Use Case Diagram*
- *Class Diagram (Customer, Appointment, Employee, Service, Admin classes)*

Appendix C: To Be Determined List

- *TBD-1: Mobile UI design*

- *TBD-2: Email notification template approval*

Alternative User Requirements

1. ***Online Payment Integration:*** *Customers should be able to pay during booking.*
2. ***Loyalty Points System:*** *Customers earn points after each appointment.*
3. ***Multi-Branch Support:*** *Multiple salon branches can be managed in one system.*
4. ***AI-Based Slot Suggestion:*** *The system recommends optimal time slots based on load.*
5. ***Customer Feedback Form:*** *Customers can evaluate their service after appointments.*