Software Requirement Specification (SRS)

Project: Queue Management System (Web & Mobile Ready)
Version: 1.0
Author: Nischay Pandey
Date: 22 / 09 / 2025

---

1. Introduction

1.1 Background

Many organizations like hospitals, banks, colleges, and service centers face long queues, wasting users' time and creating stress.
This **Queue Management System (QMS)** is a modern solution that allows users to book appointments and get services without waiting in physical lines.
It will be accessible via **web browsers and mobile devices**, and will also support **future mobile apps** by using a backend with REST APIs.

1.2 Purpose

The QMS aims to:

- Reduce waiting time and energy spent in queues.
- Provide fair and organized service management.
- Enable service providers to monitor, manage, and optimize queues efficiently.
- Allow users to book appointments online with real-time updates.

1.3 Scope

The system will include:

- **Multi-role support:** Super Admin, Branch/Admin, Users.
- **Service & Branch management:** Users can select services and see all connected branches.
- **Appointment booking:** Users can select available slots with service-specific details.
- **Real-time queue updates & notifications** through web, SMS, or email.
- **Analytics and reporting** for admins.
- **Mobile-ready backend APIs** for future app development.

---

2. Objectives

1. Replace manual queues with digital booking.
2. Provide priority handling (senior citizens, children, emergency cases).
3. Generate tickets with QR codes and time slots.
4. Allow users to reschedule or cancel bookings.
5. Provide **role-based dashboards** (Super Admin, Branch Admin, User).
6. Enable real-time notifications and queue updates.
7. Make the system cross-platform (web now, mobile later).
8. Ensure secure authentication, authorization, and data storage.

---

3. User Roles & Features

3.1 Super Admin

- Manage all service categories and branches.
- Approve or reject branch admin registrations.
- View statistics: active queues, busiest services, branch performance.
- Receive notifications for important events.
- Generate global reports.

3.2 Branch/Service Admin

- Register branch and service details (name, location, service types).
- Wait for Super Admin approval.
- Manage branch appointments and queues.
- Define service slots, working hours, ~~breaks, and holidays~~.
- Notify users of queue updates and upcoming appointments.
- Generate branch-specific reports.

3.3 User

- Register and manage profile details.
- Select service category → choose branch → view available slots.

- Book appointments and receive **QR ticket**.
- Track live queue status in real-time.
- Receive notifications for upcoming slots.
- Reschedule or cancel appointments.

---

4. Functional Requirements

4.1 User

- Registration/Login.
- Select service category & branch.
- View branch services and available slots.
- Book appointment and receive confirmation (QR code).
- Live queue tracking.
- Notifications (web push / SMS / email).
- View appointment history & reschedule.

4.2 Branch Admin

- Branch registration & approval by Super Admin.
- Define service slots, ~~breaks, and holidays.~~ Remove Holiday
- View & manage appointments/queues.
- Notify users of slot changes.
- Generate branch-specific daily/monthly reports.

4.3 Super Admin

- Add/edit/delete services and branches.
- Approve/reject branch admins.
- Monitor all queues in real-time.
- Receive alerts for important events.
- Generate global analytics reports.

4.4 Notifications & Real-Time

- Notify users when next in queue.
- Alert admins for urgent tasks or updates.
- Live queue status display.

- Use ✓ **Socket.IO / Firebase** for web real-time updates.

---

5. Non-Functional Requirements

- **Performance:** Handle multiple concurrent users.
- **Scalability:** Add more services, branches, and users without issues.
- **Security:** Role-based access, encrypted ✗ data, secure login.
- **Usability:** Responsive and intuitive UI using Tailwind CSS.
- **Cross-Platform:** Works on web now, mobile later.
- **Reliability:** 24/7 operation with minimal downtime.

---

6. Technology Stack

Layer	Technology
Frontend	HTML, Tailwind CSS, JavaScript / TypeScript
Backend	Node.js + Express (REST APIs), Socket.IO for real-time
Database	MySQL / PostgreSQL
Notifications	Firebase, Twilio (SMS/Email)
✗ Mobile-ready	APIs compatible with future React Native / Flutter app

---

7. Database Schema (High-Level)

- **Users:** id, name, [redacted], email, phone, role
- **SuperAdmin:** id, credentials
- **BranchAdmin:** id, branch_id, credentials, approved_status
- **Services:** id, name, description
- **Branches:** id, service_id, admin_id, address, timing, slots
- **Appointments:** id, user_id, branch_id, slot_time, status, QR_code
- **Notifications:** id, target_user_id/admin_id, message, read_status
- **PrioritySettings:** id, criteria (senior, children, emergency)

---

8. Use Case Overview

User

1. Register/Login
2. Select service → choose branch → view available slots
3. Book appointment → receive QR code
4. Track live queue → get notifications
5. Reschedule or cancel if needed

Branch Admin

1. Register branch → wait for Super Admin approval
2. Define services, slots, ~~breaks, holidays~~
3. View & manage queue → notify users
4. Generate branch reports

Super Admin

1. Manage services & branches
2. Approve/reject branch admins
3. Monitor global queues
4. Generate analytics & reports

---

9. Benefits

For Users

- Save time, no long queues.
- Real-time queue tracking.
- Notifications for upcoming slots.
- ~~Fair system with priority handling~~.

For Admins & Super Admin

- Organized queue management.
- Real-time monitoring and reporting.
- Efficient resource allocation.

- ❌ Mobile-ready backend for future app integration.

---

10. Conclusion

This **enhanced Queue Management System** combines traditional queue management, real-time updates, multi-role dashboards, service/branch management, a[redacted]. It ensures **efficient, fair, and organized service delivery** for both users and service providers.