

Software Requirements Specification

for

Session Booking Platform

Garv Kundnani

September 10, 2025

IEEE Standard 830-1998

Version 1.0
September 10, 2025

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Document Conventions	4
1.3	Intended Audience and Reading Suggestions	4
1.4	Product Scope	4
1.5	References	5
2	Overall Description	5
2.1	Product Perspective	5
2.2	Product Functions	5
2.2.1	Client Functions	5
2.2.2	Administrative Functions	6
2.3	User Classes and Characteristics	6
2.3.1	Clients	6
2.3.2	Administrators	6
2.4	Operating Environment	6
2.4.1	Client-Side Environment	6
2.4.2	Server-Side Environment	6
2.5	Design and Implementation Constraints	7
2.6	User Documentation	7
2.7	Assumptions and Dependencies	7
2.7.1	Assumptions	7
2.7.2	Dependencies	7
3	External Interface Requirements	7
3.1	User Interfaces	7
3.1.1	Client Interface	8
3.1.2	Administrative Interface	8
3.2	Hardware Interfaces	8
3.3	Software Interfaces	8
3.3.1	Database Interface	8
3.3.2	External Service Interfaces	8
3.4	Communications Interfaces	8
4	System Features	9
4.1	User Authentication and Authorization	9
4.1.1	Description and Priority	9
4.1.2	Stimulus/Response Sequences	9
4.1.3	Functional Requirements	9
4.2	Session Booking Management	9
4.2.1	Description and Priority	9
4.2.2	Stimulus/Response Sequences	9
4.2.3	Functional Requirements	10
4.3	Package Management	10
4.3.1	Description and Priority	10
4.3.2	Stimulus/Response Sequences	10
4.3.3	Functional Requirements	10

4.4	Administrative Dashboard	10
4.4.1	Description and Priority	10
4.4.2	Stimulus/Response Sequences	11
4.4.3	Functional Requirements	11
4.5	Program Management	11
4.5.1	Description and Priority	11
4.5.2	Stimulus/Response Sequences	11
4.5.3	Functional Requirements	11
4.6	Notification System	12
4.6.1	Description and Priority	12
4.6.2	Stimulus/Response Sequences	12
4.6.3	Functional Requirements	12
5	Other Non-functional Requirements	12
5.1	Performance Requirements	12
5.2	Safety Requirements	12
5.3	Security Requirements	13
5.4	Software Quality Attributes	13
5.4.1	Reliability	13
5.4.2	Availability	13
5.4.3	Maintainability	13
5.4.4	Portability	13
5.5	Business Rules	13
6	Other Requirements	14
6.1	Database Requirements	14
6.2	Internationalization Requirements	14
6.3	Legal Requirements	14
Appendix A: Glossary		14
Appendix B: To Be Determined List		15

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for the Session Booking Platform developed for CoreFlex Pilates Studio. This document is intended for the development team, project stakeholders, and quality assurance personnel.

The Session Booking Platform is a comprehensive web-based application that enables clients to book fitness sessions, manage their membership packages, and interact with the studio's services. It also provides administrative capabilities for studio management.

1.2 Document Conventions

This document follows the IEEE Standard 830-1998 for Software Requirements Specifications. The requirements are prioritized using the following conventions:

- **High Priority:** Critical requirements that must be implemented
- **Medium Priority:** Important requirements that should be implemented
- **Low Priority:** Desirable requirements that may be implemented

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Software developers and architects
- Project managers
- Quality assurance personnel
- CoreFlex Studio administrators
- System integrators

It is recommended to read the entire document sequentially for first-time readers. Developers should focus on Sections 3 and 4, while project managers may concentrate on Sections 1, 2, and 5.

1.4 Product Scope

The Session Booking Platform is designed to digitize and streamline the session booking process for CoreFlex Pilates Studio. The system encompasses:

- Client registration and authentication
- Session booking and management
- Package management and tracking
- Administrative dashboard for studio management
- Real-time session availability tracking

- Automated notification system

The platform aims to enhance the customer experience while providing efficient management tools for studio operations.

1.5 References

- IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Next.js Documentation: <https://nextjs.org/docs>
- MongoDB Documentation: <https://docs.mongodb.com/>
- React Documentation: <https://reactjs.org/docs/>

2 Overall Description

2.1 Product Perspective

The Session Booking Platform is a standalone web application built using modern web technologies. The system architecture consists of:

- **Frontend:** React-based web application using Next.js framework
- **Backend:** RESTful API built with Next.js API routes
- **Database:** MongoDB for data persistence
- **Authentication:** JWT-based authentication with OTP verification
- **Deployment:** Cloud-based deployment supporting web browsers

The system integrates with external services including email providers for notifications and SMS services for OTP delivery.

2.2 Product Functions

The major functions of the Session Booking Platform include:

2.2.1 Client Functions

- User registration and profile management
- OTP-based secure authentication
- Browse available programs (Pilates, Yoga, Aerial Yoga, etc.)
- View real-time session availability
- Book and cancel sessions
- View upcoming sessions and booking history
- Manage active membership packages
- Track remaining sessions in packages

2.2.2 Administrative Functions

- Admin dashboard with comprehensive analytics
- Client management and package administration
- Session scheduling and capacity management
- Program management (add, modify, delete programs)
- Generate reports on bookings and utilization
- Manage studio calendar and availability

2.3 User Classes and Characteristics

The system supports two primary user classes:

2.3.1 Clients

- **Characteristics:** End users seeking to book fitness sessions
- **Technical Expertise:** Basic to intermediate computer/smartphone usage
- **Frequency of Use:** Regular usage for booking and managing sessions
- **Access Level:** Limited to personal data and booking functions

2.3.2 Administrators

- **Characteristics:** Studio staff and management personnel
- **Technical Expertise:** Intermediate to advanced system usage
- **Frequency of Use:** Daily administrative tasks
- **Access Level:** Full system access including client management

2.4 Operating Environment

2.4.1 Client-Side Environment

- **Web Browsers:** Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- **Mobile Devices:** iOS 13+, Android 8.0+
- **Network:** Internet connectivity required
- **Screen Resolution:** Responsive design supporting 320px to 4K displays

2.4.2 Server-Side Environment

- **Runtime:** Node.js 18.0 or higher
- **Database:** MongoDB 5.0 or higher
- **Framework:** Next.js 15.x
- **Deployment:** Cloud hosting platform (Vercel, AWS, etc.)

2.5 Design and Implementation Constraints

- Must be responsive and mobile-friendly
- Must comply with web accessibility standards (WCAG 2.1)
- Authentication must use secure OTP-based verification
- All sensitive data must be encrypted in transit and at rest
- System must support concurrent users without performance degradation
- Must integrate with existing studio communication channels

2.6 User Documentation

The system will include:

- User manual for clients
- Administrator guide
- API documentation for developers
- Installation and deployment guide
- Troubleshooting guide

2.7 Assumptions and Dependencies

2.7.1 Assumptions

- Users have access to email for OTP verification
- Stable internet connectivity is available
- Users have basic familiarity with web applications
- Studio operates during defined business hours

2.7.2 Dependencies

- MongoDB database service availability
- Email service provider for notifications
- SMS service for OTP delivery
- Cloud hosting platform reliability
- Third-party library updates and security patches

3 External Interface Requirements

3.1 User Interfaces

The system provides intuitive web-based interfaces optimized for both desktop and mobile devices.

3.1.1 Client Interface

- **Home Page:** Studio information, program showcase, contact details
- **Authentication:** Login/register forms with OTP verification
- **Dashboard:** Personal session overview and quick booking access
- **Booking Interface:** Program selection, date/time picker, confirmation
- **Profile Management:** Account settings and package information
- **Session History:** Past and upcoming session management

3.1.2 Administrative Interface

- **Admin Dashboard:** System overview, analytics, and quick actions
- **Client Management:** User accounts, packages, and session tracking
- **Session Management:** Schedule creation, capacity management
- **Program Management:** Program configuration and maintenance
- **Reports:** Booking analytics and utilization reports

3.2 Hardware Interfaces

The system operates as a web application and does not require specific hardware interfaces beyond standard web browser compatibility.

3.3 Software Interfaces

3.3.1 Database Interface

- **Database Management System:** MongoDB 5.0+
- **Connection:** Mongoose ODM for database operations
- **Data Format:** BSON documents with JSON-like structure

3.3.2 External Service Interfaces

- **Email Service:** SMTP integration for notifications
- **Authentication Service:** JWT token management
- **File Storage:** Cloud-based asset storage for images

3.4 Communications Interfaces

- **HTTP/HTTPS:** RESTful API communication
- **WebSocket:** Real-time updates for session availability
- **Email Protocols:** SMTP for automated notifications
- **JSON:** Data exchange format

4 System Features

4.1 User Authentication and Authorization

4.1.1 Description and Priority

High priority feature providing secure access control through OTP-based authentication.

4.1.2 Stimulus/Response Sequences

1. User enters email address
2. System generates and sends OTP
3. User enters received OTP
4. System validates OTP and creates session
5. User gains access to authenticated features

4.1.3 Functional Requirements

- REQ-1.1: System shall support email-based user registration
- REQ-1.2: System shall generate secure 6-digit OTP codes
- REQ-1.3: System shall expire OTP codes after 10 minutes
- REQ-1.4: System shall support role-based access (client/admin)
- REQ-1.5: System shall maintain user sessions using JWT tokens
- REQ-1.6: System shall automatically logout users after 15 days of inactivity

4.2 Session Booking Management

4.2.1 Description and Priority

High priority feature enabling clients to book, modify, and cancel fitness sessions.

4.2.2 Stimulus/Response Sequences

1. Client selects program type
2. System displays available time slots
3. Client selects preferred slot
4. System checks availability and package validity
5. System confirms booking and updates availability
6. Client receives booking confirmation

4.2.3 Functional Requirements

- REQ-2.1: System shall display real-time session availability
- REQ-2.2: System shall prevent overbooking beyond session capacity
- REQ-2.3: System shall validate user package before booking
- REQ-2.4: System shall deduct sessions from user packages upon booking
- REQ-2.5: System shall allow booking cancellation up to 2 hours before session
- REQ-2.6: System shall restore session credits for valid cancellations
- REQ-2.7: System shall send confirmation notifications via email

4.3 Package Management

4.3.1 Description and Priority

High priority feature for managing client membership packages and session credits.

4.3.2 Stimulus/Response Sequences

1. Admin creates package for client
2. System assigns sessions and validity period
3. Client views package details in dashboard
4. System tracks session usage and expiration
5. System notifies clients of low session counts

4.3.3 Functional Requirements

- REQ-3.1: System shall support multiple active packages per user
- REQ-3.2: System shall track package start and end dates
- REQ-3.3: System shall monitor remaining session counts
- REQ-3.4: System shall prevent booking when no valid packages exist
- REQ-3.5: System shall notify users when packages are near expiration
- REQ-3.6: System shall support package extensions and modifications

4.4 Administrative Dashboard

4.4.1 Description and Priority

High priority feature providing comprehensive management capabilities for studio administrators.

4.4.2 Stimulus/Response Sequences

1. Administrator logs into admin dashboard
2. System displays overview of current bookings and statistics
3. Administrator selects management function
4. System provides detailed interface for selected function
5. Administrator performs required actions
6. System updates data and provides confirmation

4.4.3 Functional Requirements

- REQ-4.1: System shall provide client management interface
- REQ-4.2: System shall enable session schedule management
- REQ-4.3: System shall support program creation and modification
- REQ-4.4: System shall generate booking and utilization reports
- REQ-4.5: System shall provide real-time booking notifications
- REQ-4.6: System shall allow bulk operations on sessions and clients

4.5 Program Management

4.5.1 Description and Priority

Medium priority feature for managing different types of fitness programs offered.

4.5.2 Stimulus/Response Sequences

1. Administrator accesses program management
2. System displays current programs
3. Administrator creates, modifies, or deletes programs
4. System validates program information
5. System updates program database
6. System reflects changes in booking interface

4.5.3 Functional Requirements

- REQ-5.1: System shall support multiple program types (Pilates, Yoga, etc.)
- REQ-5.2: System shall allow program descriptions and details
- REQ-5.3: System shall enable program activation and deactivation
- REQ-5.4: System shall maintain program booking history
- REQ-5.5: System shall support program-specific booking rules

4.6 Notification System

4.6.1 Description and Priority

Medium priority feature for automated communication with users.

4.6.2 Stimulus/Response Sequences

1. System event triggers notification requirement
2. System determines appropriate notification method
3. System generates personalized notification content
4. System sends notification via configured channel
5. System logs notification delivery status

4.6.3 Functional Requirements

- REQ-6.1: System shall send booking confirmation emails
- REQ-6.2: System shall send session reminder notifications
- REQ-6.3: System shall notify about package expiration warnings
- REQ-6.4: System shall send cancellation confirmations
- REQ-6.5: System shall provide notification preferences management

5 Other Non-functional Requirements

5.1 Performance Requirements

- PERF-1: System shall respond to user requests within 2 seconds under normal load
- PERF-2: System shall support minimum 100 concurrent users
- PERF-3: Database queries shall execute within 1 second for 95% of requests
- PERF-4: System shall handle peak booking periods with less than 5 seconds response time
- PERF-5: System uptime shall be at least 99.5% during business hours

5.2 Safety Requirements

- SAFE-1: System shall prevent data loss through regular automated backups
- SAFE-2: System shall maintain data integrity during concurrent operations
- SAFE-3: System shall implement proper error handling to prevent system crashes
- SAFE-4: System shall validate all user inputs to prevent injection attacks
- SAFE-5: System shall log all critical operations for audit purposes

5.3 Security Requirements

- SEC-1: All data transmission shall use HTTPS encryption
- SEC-2: User passwords shall be hashed using industry-standard algorithms
- SEC-3: System shall implement rate limiting to prevent abuse
- SEC-4: Session tokens shall expire after defined periods
- SEC-5: System shall log all authentication attempts
- SEC-6: Administrative functions shall require additional authentication

5.4 Software Quality Attributes

5.4.1 Reliability

- The system shall maintain 99.5% uptime during business hours
- Mean Time Between Failures (MTBF) shall exceed 720 hours
- System shall gracefully handle and recover from errors

5.4.2 Availability

- System shall be available 24/7 with scheduled maintenance windows
- Maintenance windows shall not exceed 4 hours monthly
- System shall provide status page for service availability

5.4.3 Maintainability

- Code shall follow established coding standards and documentation
- System shall support modular architecture for easy updates
- Database schema shall support non-destructive migrations

5.4.4 Portability

- System shall run on multiple cloud platforms
- Database shall be platform-independent
- System shall support horizontal scaling

5.5 Business Rules

- BR-1: Clients can only book sessions if they have valid active packages
- BR-2: Session cancellations must be made at least 2 hours in advance
- BR-3: Each session has a maximum capacity that cannot be exceeded
- BR-4: Packages have specific validity periods and cannot be used after expiration
- BR-5: Only administrators can create, modify, or delete programs

- BR-6: Session credits are deducted immediately upon booking confirmation
- BR-7: Refunded sessions are restored to the user's package balance

6 Other Requirements

6.1 Database Requirements

- The system shall use MongoDB as the primary database
- Database shall maintain ACID properties for critical operations
- Data shall be backed up daily with point-in-time recovery capability
- Database shall support indexing for optimal query performance
- Database schema shall accommodate future feature expansions

6.2 Internationalization Requirements

- System shall support English language interface
- System shall accommodate Indian timezone (IST) for all operations
- Date and time formats shall follow local conventions
- Currency display shall support Indian Rupee (INR)

6.3 Legal Requirements

- System shall comply with Indian data protection regulations
- User consent shall be obtained for data processing
- System shall provide data export functionality for user requests
- System shall support data deletion upon user request

Appendix A: Glossary

Term	Definition
API	Application Programming Interface - a set of protocols and tools for building software applications
JWT	JSON Web Token - a compact, URL-safe means of representing claims between two parties
MongoDB	A document-based NoSQL database system
Next.js	A React-based web development framework
OTP	One-Time Password - a password that is valid for only one login session
REST	Representational State Transfer - an architectural style for distributed systems
SRS	Software Requirements Specification

Package	A membership bundle containing a specific number of sessions with validity period
Session	A scheduled fitness class with defined start time, duration, and capacity
Slot	A time period during which a session can be scheduled

Appendix B: To Be Determined List

- Integration with payment gateway for online payments
- Mobile application development timeline
- Advanced reporting and analytics features
- Integration with third-party calendar systems
- Multi-language support implementation
- Automated waitlist management system