PRD Sample: Online Booking & Scheduling Platform

SlotWise: A venue and activity booking web application

2026

SlotWise: Booking & Scheduling Platform

Document type: Product Requirement Document (Sample) **Version:** 1.0 **Scope:** Customer-facing web app + Admin dashboard

---

1. Overview

SlotWise is an online booking platform for venues and activity centres. Customers browse available time slots, make a booking, and pay online. Venue administrators manage availability, view reservations, and track revenue from a dashboard.

This document covers the web application only. A companion mobile app is out of scope for this version.

---

2. User Personas

Customer (end user) A person booking an activity slot: searching for available times, selecting a slot, paying, and receiving confirmation. Accesses the platform via any device on a browser.

Venue Admin The business owner or staff member who manages the venue's schedule, reviews upcoming bookings, creates manual bookings for walk-in customers, and monitors revenue.

Super Admin The platform operator who manages multiple venues, users, and platform-wide settings. Out of scope for v1.

---

3. Data Models

3.1 Venue

Field	Type	Notes
`id`	integer	Auto-increment primary key
`name`	string	Public name of the venue
`slug`	string	URL-safe unique identifier

Field	Type	Notes
description	text	Shown on the venue listing page
address	string	Physical address
timezone	string	IANA timezone e.g. Asia/Kolkata
cover_image	file	Hero image for the venue page
is_active	boolean	Controls whether the venue accepts bookings
created_at	timestamp	
updated_at	timestamp	

3.2 Activity

Field	Type	Notes
id	integer	
venue_id	integer	Foreign key to Venue
name	string	e.g. “Bay 1: Full Hour”
description	text	
duration_minutes	integer	Length of a single slot
price	decimal	Base price per slot (before tax)
max_capacity	integer	Maximum number of participants per slot
is_active	boolean	

3.3 Availability Rule

Field	Type	Notes
id	integer	
activity_id	integer	Foreign key to Activity
day_of_week	integer	0 = Sunday … 6 = Saturday
open_time	time	Start of bookable window e.g. 09:00
close_time	time	End of bookable window e.g. 22:00
slot_interval_minutes	integer	Gap between slot start times e.g. 30

3.4 Booking

Field	Type	Notes
id	integer	
booking_reference	string	Human-readable e.g. SW-2026-00147
activity_id	integer	
customer_name	string	
customer_email	string	

Field	Type	Notes
`customer_phone`	string	
`participant_count`	integer	
`slot_start_at`	datetime	UTC timestamp of slot start
`slot_end_at`	datetime	UTC timestamp of slot end
`status`	enum	`pending`, `confirmed`, `cancelled`, `no_show`
`payment_status`	enum	`unpaid`, `paid`, `refunded`
`payment_reference`	string	Gateway transaction ID
`amount_paid`	decimal	Total charged including tax
`coupon_id`	integer	Nullable; applied coupon
`notes`	text	Admin-visible notes
`booking_method`	enum	`online`, `manual`
`created_at`	timestamp	

3.5 Coupon

Field	Type	Notes
`id`	integer	
`code`	string	Case-insensitive; unique
`discount_type`	enum	`flat` or `percent`
`discount_value`	decimal	Amount or percentage off
`max_uses`	integer	Nullable = unlimited
`use_count`	integer	Incremented on each successful application
`valid_from`	date	
`valid_until`	date	Nullable = no expiry
`is_active`	boolean	

---

4. Features

Feature 1: Browse & Search

US-01: View activity listings As a customer, I can view all available activities for a venue so that I can decide what to book.

Acceptance criteria:

1. The venue page shows all active activities with name, duration, price, and a “Book Now” button.
2. Activities with `is_active = false` do not appear.
3. The page is responsive and usable on mobile viewports (min 360px width).

US-02: Select a date and see available slots As a customer, I can pick a date and see all open time slots for an activity so that I can find a time that suits me.

Acceptance criteria:

1. A date picker is shown after the customer selects an activity. Only today and future dates are selectable.
2. Available slots for the selected date are derived from the Availability Rule for the activity's day of week.
3. Slots where an existing confirmed booking has reached `max_capacity` are shown as "Full" and cannot be selected.
4. Slots in the past (relative to the venue's timezone) are shown as disabled even if today is selected.
5. If no slots are available for a date, the message "No slots available on this date" is shown.

Feature 2: Booking Flow

US-03: Enter booking details As a customer, I can enter my name, email, phone number, and number of participants so that my reservation is personalised.

Acceptance criteria:

1. The form requires: name (text, required), email (valid email format, required), phone (required, minimum 7 digits), participants (integer, min 1, max = activity `max_capacity`).
2. Validation errors are shown inline beneath each field on submit attempt.
3. Submitting a valid form proceeds to payment. No booking record is created at this step.

US-04: Apply a coupon As a customer, I can enter a coupon code to reduce the booking cost so that I can use promotions I have received.

Acceptance criteria:

1. An optional coupon field is shown on the booking details step.
2. On entry, the code is validated in real time (debounced, 500ms). If valid and active, the discount is applied and the new total is shown.
3. If the code is invalid, expired, or exhausted, an inline error is shown and the full price remains.
4. Only one coupon can be applied per booking.

US-05: Pay online As a customer, I can complete payment via the integrated payment gateway so that my booking is confirmed immediately.

Acceptance criteria:

1. Payment is processed via the configured gateway (Razorpay or Stripe: to be specified at project start).
2. On successful payment, a booking record is created with `status = confirmed` and `payment_status = paid`.
3. The customer is redirected to a confirmation page showing: booking reference, activity name, date/time, participant count, and amount paid.
4. A confirmation email is sent to the customer's email address within 60 seconds of payment completion.
5. On payment failure the customer is returned to the payment step with an error message. No booking is created.

US-06: Receive booking confirmation email As a customer, I receive an email confirming my booking details so that I have a record without needing to log in.

Acceptance criteria:

1. Email contains: booking reference, venue name, activity name, date and time (in venue timezone), participant count, total amount paid, and a cancellation link.
2. Email is sent from a verified sender domain. It must not require the customer to create an account.

Feature 3: Admin Dashboard

US-07: View all bookings As an admin, I can view all bookings for my venue so that I can manage operations.

Acceptance criteria:

1. The bookings list shows: booking reference, customer name, activity, slot date/time, participants, amount, status, booking method.
2. The list is paginated (25 rows per page, default).
3. Filters available: date range, activity, status (confirmed / cancelled / no_show), booking method (online / manual).
4. A search field filters by booking reference or customer name.

US-08: Create a manual booking As an admin, I can create a booking directly from the dashboard for walk-in or phone customers so that all bookings are tracked in one place.

Acceptance criteria:

1. A "New Booking" button opens a form with the same fields as the customer flow plus a "Payment method" field (Cash / Card / UPI / Waived).
2. The slot availability check runs on form submission. If the slot is full, an error is shown.
3. Created bookings have `booking_method = manual`. If "Waived" is selected, `payment_status = paid` and `amount_paid = 0`.
4. A confirmation email is sent to the customer if an email address is provided.

US-09: Cancel a booking and issue a refund As an admin, I can cancel a confirmed booking and trigger a refund so that customers are not charged for cancelled slots.

Acceptance criteria:

1. Cancelling a booking sets `status = cancelled`.
2. If `payment_status = paid` and `booking_method = online`, the admin is prompted to choose: "Refund via gateway" or "Mark refunded manually".
3. "Refund via gateway" calls the payment gateway's refund API. On success, `payment_status = refunded`.
4. The cancelled slot becomes available again for new bookings immediately.

5. A cancellation notification email is sent to the customer.

US-10: View revenue analytics As an admin, I can view revenue and booking summaries so that I can understand my business performance.

Acceptance criteria:

1. The analytics dashboard shows for a selectable date range: total bookings, total revenue, cancelled bookings, bookings by activity.
2. A chart displays daily revenue over the selected period.
3. A "Busiest slots" table shows the top 5 time slots by booking count for the period.
4. Data is exportable as a CSV file.

---

5. Scope Boundaries

In scope: - Customer booking flow (browse ☒ select slot ☒ enter details ☒ pay ☒ confirm) - Online payment via one gateway (to be agreed at project start) - Coupon / discount codes - Admin booking management (create, view, cancel, refund) - Admin revenue dashboard - Confirmation and cancellation emails

Out of scope for v1: - Customer login or booking history (guest checkout only) - Multi-venue management - Recurring / subscription bookings - SMS notifications - Mobile apps - Third-party calendar sync (e.g. Google Calendar)

---

6. Prerequisites

The following must be provided before development begins:

1. UI designs (Figma or equivalent) for all customer-facing screens and the admin dashboard.
2. Decision on payment gateway (Razorpay or Stripe) and API credentials.
3. Brand assets: logo, colour palette, fonts.
4. Email sender domain and SMTP / transactional email provider credentials.
5. Agreed hosting environment (server or cloud provider).

This is a sample document demonstrating the level of detail expected in a Product Requirement Document. Names, figures, and features are illustrative.