SMARTPARK V2 – Vehicle Parking Management System
Use Case Specifications

---

1. Use Case Specification

Field	Details
Use Case ID	UC-001
Use Case Name	Register as Parker or Parking Host
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last 2Updated	-
Actors	Parker, Parking Host
Description	A new user creates an account by choosing their role (Parker or Parking Host) to access the platform features.
Preconditions	User is not previously registered.
Post conditions	User account is created with selected role and stored in the database; user is redirected to login page.
Normal Flow	1. User navigates to Registration page. 2. Selects role (Parker / Parking Host). 3. Enters username, email, password, full name, phone (optional). 4. System validates data. 5. Account is created successfully.
Alternate Flows	A1: User clicks “Already have an account” → redirects to Login (UC-002).
Exceptions	E1: Email/username already exists. E2: Passwords do not match.
Priority	High
Frequency of Use	Medium
Business Rules	Email must be unique; password must be ≥ 8 characters with uppercase, number & special character.

Assumptions	Email verification is optional for local demo.

---

2. Use Case Specification

Field	Details
Use Case ID	UC-002
Use Case Name	Login to the System
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Super Admin, Parking Host, Parker
Description	Authenticated user logs into the system based on their role.
Preconditions	User has a valid registered account.
Post conditions	User is logged in with JWT/session and redirected to role-specific dashboard.
Normal Flow	1. User enters username/email and password. 2. System validates credentials. 3. Role is checked. 4. User is redirected to appropriate dashboard.
Exceptions	E1: Invalid credentials. E2: Account locked after multiple failed attempts.
Priority	High
Frequency of Use	High
Business Rules	Role-based access control enforced. Super Admin is pre-seeded.
Assumptions	JWT or Flask-Login is used.

---

3. Use Case Specification

Field	Details

Use Case ID	UC-003
Use Case Name	Create Public Parking Lot (Super Admin)
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Super Admin
Description	Super Admin creates a new public parking lot and system auto-generates the requested number of parking spots.
Preconditions	Super Admin is logged in.
Post conditions	Parking lot and spots are created in database.
Normal Flow	1. Admin clicks “Create Parking Lot”. 2. Enters name, address, pincode, price per hour, total spots. 3. Submits form. 4. System creates lot + auto-generates spots (all available).
Exceptions	E1: Duplicate lot name in same area.
Priority	High
Frequency of Use	Medium
Business Rules	Lot can be deleted only if all spots are empty.
Assumptions	Spots are auto-created programmatically.

---

4. Use Case Specification

Field	Details
Use Case ID	UC-004
Use Case Name	List Private Parking Spot (Parking Host)
Created By	Development Team
Last Updated By	Development Team

Date Created	-
Date Last Updated	-
Actors	Parking Host
Description	Parking Host lists a private spot (house/store driveway) for monetization.
Preconditions	Host is logged in.
Post conditions	Private spot is published and visible to Parkers.
Normal Flow	1. Host goes to “My Spots” → “List New Spot”. 2. Enters address, pincode, price/hour, description, availability. 3. Submits → spot becomes active.
Exceptions	E1: Overlapping availability.
Priority	High
Frequency of Use	Medium
Business Rules	Host earns tracked revenue from bookings.
Assumptions	Photo upload is simulated via URLs.

---

5. Use Case Specification

Field	Details
Use Case ID	UC-005
Use Case Name	Search & Reserve Parking Spot for Time Slot
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Parker
Description	Parker searches and books a public or private spot for a specific time slot.
Preconditions	Parker is logged in and spots are available.

Post conditions	Reservation is created; spot is reserved for the selected time.
Normal Flow	1. Parker enters pincode/date/time. 2. System shows available spots. 3. Parker selects spot and time slot. 4. System checks availability. 5. Parker confirms booking.
Exceptions	E1: No spots available in selected slot.
Priority	High
Frequency of Use	High
Business Rules	No overlapping bookings allowed on same spot.
Assumptions	Real-time availability uses Redis cache.

---

6. Use Case Specification

Field	Details
Use Case ID	UC-006
Use Case Name	Check-in Vehicle (Parker)
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Parker
Description	Parker marks the vehicle as parked at the allotted spot.
Preconditions	Valid active reservation exists.
Post conditions	Spot status changes to occupied; parking timestamp recorded.
Normal Flow	1. Parker goes to “My Active Bookings”. 2. Clicks “Check-in”. 3. Enters vehicle number (optional). 4. System updates status.
Exceptions	E1: Attempt outside allowed time window.

Priority	High
Frequency of Use	High
Business Rules	Only the booking owner can check-in.
Assumptions	Location verification is optional for demo.

---

7. Use Case Specification

Field	Details
Use Case ID	UC-007
Use Case Name	Check-out Vehicle (Parker)
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Parker
Description	Parker vacates the spot and final cost is calculated.
Preconditions	Vehicle is checked-in.
Post conditions	Spot becomes available; reservation is completed with final cost.
Normal Flow	1. Parker clicks “Check-out”. 2. System calculates actual duration × rate. 3. Updates leaving timestamp and cost. 4. Spot status = available.
Exceptions	E1: Early or late checkout warning.
Priority	High
Frequency of Use	High
Business Rules	Minimum charge = 1 hour.
Assumptions	Payment is simulated.

---

8. Use Case Specification

Field	Details
Use Case ID	UC-008
Use Case Name	Trigger CSV Export of Parking History
Created By	Development Team
Last Updated By	Development Team
Date Created	-
Date Last Updated	-
Actors	Parker, Parking Host
Description	User triggers asynchronous export of their complete parking history as CSV.
Preconditions	User is logged in and has at least one booking.
Post conditions	CSV file is generated and download link is sent via email.
Normal Flow	1. User clicks “Export History” on dashboard. 2. Celery task is queued. 3. Task generates CSV. 4. Email with download link is sent.
Exceptions	E1: No records found.
Priority	Medium
Frequency of Use	Low
Business Rules	Export runs via Celery + Redis asynchronously.
Assumptions	Email uses console backend for local demo.