Functional Design

The application is divided into below features

1. User Management

2. Parking Maintenance

3. Booking Management

4. Payment Module

**1. USER MANAGEMENT**

There are 2 types of users supported in the application - **ADMIN** and **USER**.

**ADMIN** – can create other admins and do the overall maintenance of the parking centers

**USER** – its a normal user who can primarily book a slot for the vehicle

Below functions are supported

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Function | User types | API | Comments |
| 1 | Login | No check | **/login** |  |
| 2 | Sign Up | No check | **/signup** |  |
| 3 | Create a User | ADMIN | **/createUser** | To Create other admin users |
| 4 | Change Password | ADMIN and USER | **/changePassword** |  |
| 5 | View Profile | ADMIN and USER | **/viewProfile** |  |
| 6 | Edit Profile | ADMIN and USER | **/editProfile** |  |
| 7 | Fetch All Users | ADMIN | **/getAllUsers** |  |
| 8 | Delete a User | ADMIN | **/deleteUser/{userId}** | Deletion for any user type |
| 9 | Add vehicle | ADMIN and USER | **/addVehicle** | Registration no is unique for a user |
| 10 | View Active Vehicles | ADMIN and USER | **/getActiveVehicles** |  |
| 11 | Modify Vehicle | ADMIN and USER | **/modifyVehicle** |  |
| 12 | Delete Vehicle | ADMIN and USER | **/deleteVehicle/{vehicleId}** |  |
| 13 | Logout | All Users |  | JWT token invalidation done from UI. So no API |

Apis are present in

com.parking.user.controllers.UserResource

Entities:

1. User

2. Vehicle

**2. Parking Maintenance**

This is mainly for ADMIN for the general maintenance of the Parking centers

Below functions are supported

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Function | User types | API | Comments |
| 1 | Get Active Countries | ADMIN and USER | **/countries** | Fetch all countries |
| 2 | Get Applicable Countries | ADMIN and USER | **/applicableCountries** | Fetch countries of centers which have booked slots. For searching bookings |
| 3 | Get All Parking Centers | ADMIN and USER | **/viewAllCenters/{country}** | All parking centers for a country |
| 4 | Get Parking Center Details | ADMIN | **/viewCenterDetails/{centerId}** | Fetch center details for ADMIN |
| 5 | Create A parking Center | ADMIN | **/createCenter** | Created with a particular no of slots |
| 6 | Modify Parking Center Details | ADMIN | **/modifyCenter** |  |
| 7 | Delete A Parking Center | ADMIN | **/deleteCenter** |  |
| 8 | Get All Parking Slots for a center | ADMIN | **/viewAllParkingSlots/{centerId}** | Fetch including active and deleted slots |
| 9 | Get All Available Parking Slots for a center | ADMIN and USER | **/viewAllAvailableParkingSlots/{centerId}** | Fetch only active. To be used in booking for both admin and user |
|  | Add Slots To Center | ADMIN | **/addSlots** |  |
|  | Delete A Parking Slot | ADMIN | **/deleteSlot** |  |

Resources:

com.parking.maintenance.controller

Entities used: com.parking.maintenance.entity

1. Country

2. Center

3.Slot

4.User

**2. Booking Management**

For Booking a slot, user will select a country and center and input the From Date and To Date and click on “Find Available Slots” Button. It will fetch the available slots and then he can book a particular slot for that time period.

It will still check for slot availability in the BOOK Slot action.

Below functions are supported

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Function | User types | API | Comments |
| 1 | Get All Active Bookings | ADMIN and USER | **/activeBookings/{userId}** | Fetch all active bookings for a user id |
| 2 | Get All in active bookings | ADMIN and USER | **/inActiveBookings/{userId}** | Fetch all expired and cancelled bookings for a user |
| 3 | Get All Bookings | ADMIN | **/allBookings/{userId}** | Fetches active, cancelled, expired bookings for a user. Reporting purpose for admin |
| 4 | Get Booking Details | ADMIN and USER | **/bookingDetails/{bookingId}** | Fetch booking details for a particular Booking |
| 5 | Cancel a Booking | ADMIN and USER | **/cancelBooking** | Cancel a Booking |
| 6 | Fetch all Slots avaialable for a center and a duration | ADMIN and USER | **/bookingAvailableSlots** | Input passed:  1. List of Slot Ids  2. From time  3, To Time  Returns the slots which are available for booking |
| 7 | Book a Slot | ADMIN and USER | **/bookSlot** | Book a parking slot. It will check for slot availability before booking |

Resources:

com.parking.booking.controller

Entities used: com.parking.booking.entity

1. Booking

2.User

3. Payment Module

Pending