Parking Lot

I own a multi-storey parking lot that can hold up to 'n' cars at any given point in time. Each slot is given a number starting at 1 increasing with increasing distance from the entry point in steps of one. I want to create an automated ticketing system that allows my customers to use my parking lot without human intervention.

When a car enters my parking lot, I want to have a ticket issued to the driver. The ticket issuing process includes us documenting the registration number (number plate) and the colour of the car and allocating an available parking slot to the car before actually handing over a ticket to the driver (we assume that our customers are nice enough to always park in the slots allocated to them). The customer should be allocated a parking slot which is nearest to the entry. At the exit, the customer returns the ticket which then marks the spot they were using as being available.

Expected functionalities:

1. Create a parking slot for a given number of slots. You can assume:
    a. User has marked physical slots in order of distance - taking in account parking level/slot reachability etc.
    b. All slots are equal and any car can go in any slot.
2. Park a car
    a. Input data is car registration number and color of car.
    b. Allots the nearest slot depending upon availability.
3. Vacate a slot
    a. Input data is slot number to be freed.
4. Get status of allocated slots
    a. This would get a list of occupied slots along with registration number and color of car parked in respective slot.
5. Get status of free slots
    a. This would get a list of available slots

Expectations from solution:

Necessary:

- Clean and object-oriented low-level design.
- Appropriate coding conventions and directory structure wrt language used.
- Input can be through file, sysin or test cases.
- Error handling of edge cases. Exception handling.
- Optimality of solution in terms of time complexity for various operations.

Optional:

- Facilitate multiple request getting processed in parallel without leading to inconsistent state.

Sample of expected Input/Output:

Please find below a sequence of sample input and output expected. You could simulate the same using test-cases if you do not wish to have sysin command format input.

#	Input	Output
1	`create_parking_lot 6`	`Created a parking lot with 6 slots`
2	`park KA-01-HH-1234 white`	`Allocated slot number: 1`
3	`park KA-01-HH-9999 white`	`Allocated slot number: 2`
4	`park KA-01-BB-0001 black`	`Allocated slot number: 3`
5	`park KA-01-HH-7777 red`	`Allocated slot number: 4`
6	`park KA-01-HH-2701 blue`	`Allocated slot number: 5`
7	`park KA-01-HH-3141 black`	`Allocated slot number: 6`
8	`leave 4`	`Slot number 4 is freed`

9	`status allocated`
	
	`(Tab delimited output expected)`
		`Slot No  Registration  Color`
		`1  KA-01-HH- 1234  white`
		`2  A-01-HH-9999  white`
		`3  KA-01-BB-0001  black`
		`5  KA-01-HH-2701  blue`
		`6  KA-01-HH-3141  black`

10	`status free`	`Slots` `4`
11	`park KA-01-P-333 white`	`Allocated slot number: 4`
12	`park DL-12-AA-9999 white`	`Sorry, parking lot is full`