Parking Service (Version: 1.0)

- Language: Java
- Framework: Spring Boot
- Spring Module: Web MVC (Rest)
- Skills Required: Core Java + Spring Boot + Rest API Development

I own a 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 with the time the car was parked in the lot, which then marks the slot they were used as being available. Total parking charge should be calculated as per the parking time. Charge applicable is $10 for first 2 hours and $10 for every additional hour.

We interact with the system via a simple set of commands which produce a specific output. Please take a look at the example below, which includes all the commands you need to support - they're self explanatory. The system should accept a filename as a parameter at the command prompt and read the commands from that file.

Commands

- Create parking lot of size n : create_parking_lot {capacity}
- Park a car : park {car_number}
- Remove(Unpark) car from : leave {car_number} {hours}
- Print status of parking slot : status

Input (contents of file):

```
create_parking_lot 6
park KA-01-HH-1234
park KA-01-HH-9999
park KA-01-BB-0001
park KA-01-HH-7777
park KA-01-HH-2701
park KA-01-HH-3141
leave KA-01-HH-3141 4
status
park KA-01-P-333
park DL-12-AA-9999
leave KA-01-HH-1234 4
leave KA-01-BB-0001 6
leave DL-12-AA-9999 2
park KA-09-HH-0987
park CA-09-IO-1111
park KA-09-HH-0123
status
```

Output (to STDOUT):

```
Created parking lot with 6 slots
Allocated slot number: 1
Allocated slot number: 2
Allocated slot number: 3
Allocated slot number: 4
Allocated slot number: 5
Allocated slot number: 6
Registration number KA-01-HH-
3141 with Slot Number 6 is free with Charge 30
Slot No. Registration No.
1 KA-01-HH-1234
```

2 KA-01-HH-9999

3 KA-01-BB-0001

4 KA-01-HH-7777

5 KA-01-HH-2701

Allocated slot number: 6

Sorry, parking lot is full

Registration number KA-01-HH-

1234 with Slot Number 1 is free with Charge 30

Registration number KA-01-BB-

0001 with Slot Number 3 is free with Charge 50

Registration number DL-12-AA-9999 not found

Allocated slot number: 1

Allocated slot number: 3

Sorry, parking lot is full

Slot No. Registration No.

1 KA-09-HH-0987

2 KA-01-HH-9999

3 CA-09-IO-1111

4 KA-01-HH-7777

5 KA-01-HH-2701

6 KA-01-P-333

Code Designing Rules:

1. Use SpringBoot Project with Maven. (You may use https://start.spring.io/ to create a new SpringBoot Project)
2. Use OOPs concepts as much as you can.
3. Handle all edge cases and validations.
4. Exception handling, you are free to create custom exceptions if required.
5. Keep configurations params in application.properties.
6. Try to divide the entire application into smaller chunks.
7. Instead of creating Setter and Getter by your own use lombok maven dependency.

How to execute:

Create a RestAPI which takes filename as query parameter and prints output on console.