**Parking Management System**

This system will automate the cars parking management, when a new user will arrive and try to park a car, they will ask about the car and parking level, and using this information system will automatically find and allocate the parking to the user.

**Identify and describe the user requirements and the types of data required for this project as part of the program documentation:**

Few of the user requirements are given below:

1. System should be able to add the new cars, while keeping the required parking level in mind.
2. System should be able to remove a car and calculate its total bill.
3. Display the recently added cars in the system.
4. Display the list of currently parked cars.

**Define a clear description of the Inputs and Outputs to the system together with the developed system and its functionality as part of the program documentation:**

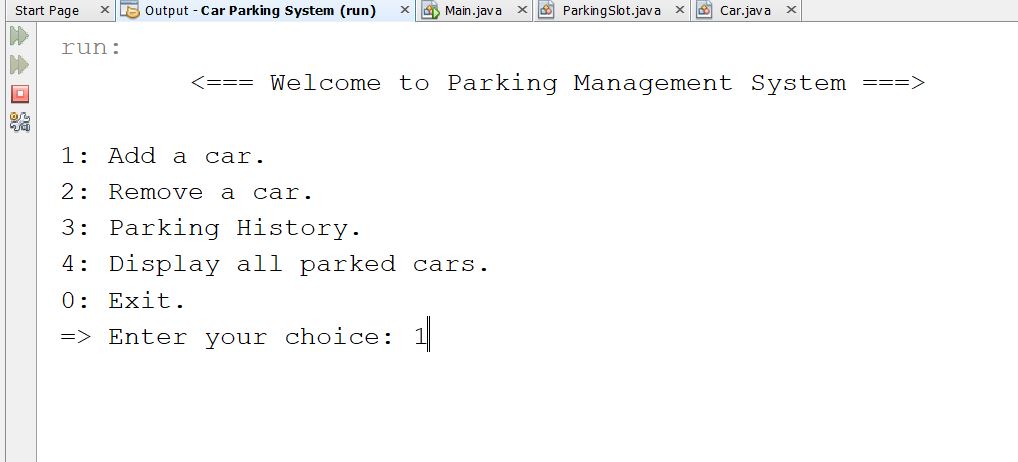
1. **User-driven menu** will ask the user to add, remove the card. Or to display the complete parking list, it will also offer to display the history of recently parked cars.
2. To **add a new car**, user will add car name, plates number, model, company, model year. Then parking level (high/normal/low). High will cost 15$ per hour, normal will cost 10$ per hour and low will cost 5$ per hour. And finally total number of hours required.
3. **Removing** a park from parking, will take only plates number. Then system will find the car and remove it from the parking. And make the parking slot available to other users.
4. **Parking history** option will display the list of recently added cars in the parking.
5. **List of all** the cars currently parked.

**Describe the methods/functions that you have created in terms of what they perform, passed in data types, and returned values and how these functions will relate to each other as part of the program documentation.**

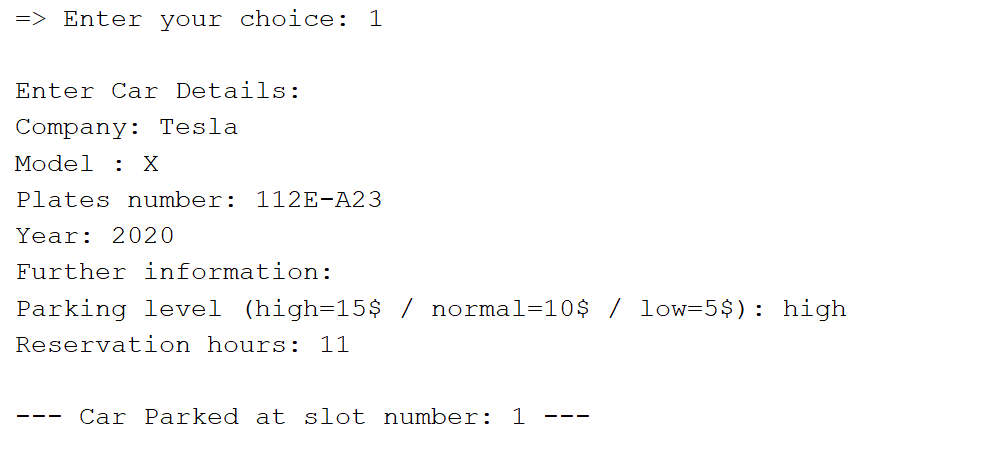
1. initializeParkingSlots(): This function will work as data initializer, It will assign the first 10 slots as high level parking, then next 10 slots to normal level and next 10 slots to low level slots.
2. addNewCar(): This function will first take the user inputs about the car and required parking. Then it will look for an available slot and assign the car to that slot. In case there is no available slot found, system will display the “No slot available message”.
3. removeCar(): This function will take the car plates number as input and search for car, after finding the car it will remove the car from the slot and make the slot available to other user. Also, it will add the removed car in the recent cars arraylist.

**Include screenshots with a simple commentary of the Application running with some sample test data:**

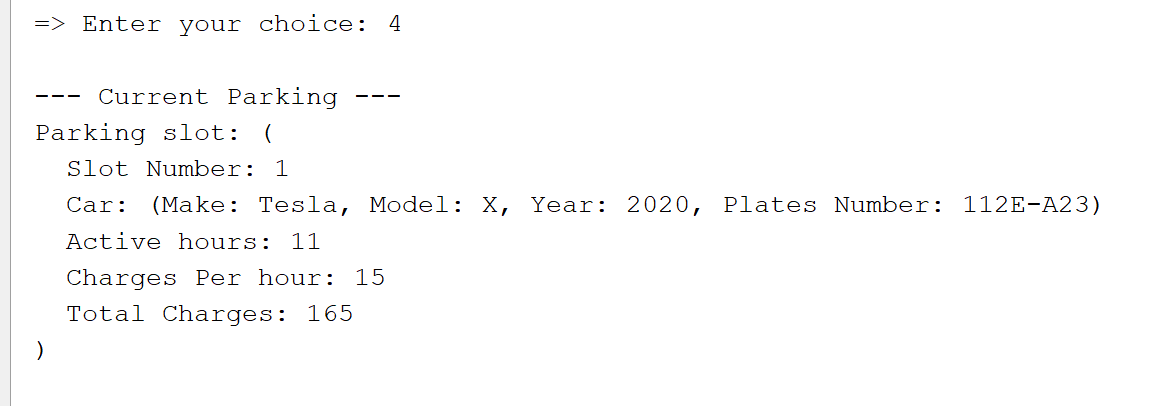
1. **User-driven menu:**



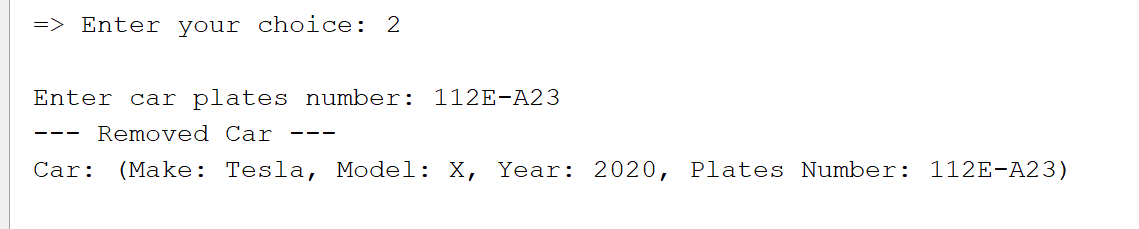
1. Add a new car:



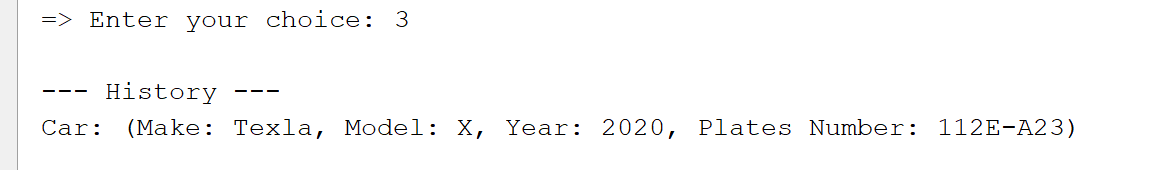
1. **Currently parked cars:**



1. **Remove a car:**



1. **History of cars:**



1. **Exit system:**

