ParkingSpotGUI Documentation

# GUI Components:

1. JPanel: parkingGridPanel: Represents the main grid where the parking slots (buttons) are displayed. The layout changes dynamically based on the current state of the CarPark (i.e., slots added or removed).
2. JTextArea: outputArea: A text area used to display messages, logs, and outputs to the user. It provides feedback on actions like parking a car, removing a car, etc.
3. JButton: exitButton: When clicked, this button exits the application.
4. JPanel: sidePanel: A side panel that possibly contains various operation buttons and controls to interact with the parking system.
5. JButton: addSlotButton: This button prompts the user to add a new parking slot. It gathers input for the slot ID, performs validations, and adds the slot if valid.
6. JButton: deleteSlotButton: When clicked, prompts the user for a parking slot ID to delete. If the slot is occupied, it won't allow deletion.
7. JPanel: buttonPanel: A panel that organizes several operation buttons, likely those related to parking and finding/removing cars.
8. JButton: showAllSlotsButton: Displays all the parking slots when clicked.
9. JButton: parkCarButton: When clicked, gathers details about the car (registration number, make, model, year) and the desired slot ID. It then parks the car in the slot if all details are valid and the slot is available.
10. JButton: findCarByRegButton: Prompts the user for a car registration number and finds the slot where the car with the given registration number is parked.
11. JButton: findCarsByMakeButton: Prompts the user for a car make and lists all the cars of that make currently parked, along with their parking slots.
12. JButton: removeCarByRegButton: Prompts the user for a car registration number and removes the car with that registration number from its parking slot.
13. JButton: createParkingSlotButton: A dynamically created button for each parking slot. Its appearance and behavior change based on whether the slot is occupied or available.
14. JPanel: findAndRemovePanel: A panel that may be organizing controls related to finding and removing cars. Needs further inspection for exact functionality.
15. JButton: button: This refers to dynamically created buttons used for parking a car in a parking slot or removing a car from the given parking slot instantaneously.
16. JButton: slotButton: A generic button related to parking slots. Its exact purpose needs further inspection.

# Event Handlers:

1. System.exit(0): Exits the application.
2. addSlotButton: Gathers slot ID from the user. Validates the input, checks for duplicates, and adds the parking slot if the ID is valid.
3. deleteSlotButton: Gathers a slot ID from the user and deletes the slot if it exists and is not occupied.
4. showAllSlotsButton: Lists all the parking slots in the outputArea.
5. parkCarButton: Gathers car details and the desired parking slot from the user. Validates the input and parks the car if all details are valid and the slot is available.
6. findCarByRegButton: Gathers a registration number from the user and displays the slot where the car with the given registration number is parked.
7. findCarsByMakeButton: Gathers a car make from the user and displays all the cars of that make currently parked, along with their slots.
8. removeCarByRegButton: Gathers a registration number from the user and removes the car with that number from its parking slot.
9. handleSlotButtonClick: Handles the click event on a parking slot button. If the slot is occupied, it displays the car details and provides an option to remove the car.
10. createParkingSlotButton: Creates a button for a parking slot and assigns an action listener to it which calls the handleSlotButtonClick method when clicked.