**Main Menu Class**

**GUI Components:**

* JFrame: This represents the main window of the application.
* JPanel:
  + contentPane: It serves as the main container for other GUI components.
  + mainPanel: It is the panel where different panels will be added based on menu selection.
  + sidePanel: It is the panel that contains the menu items.

**Event Handling Functions:**

ActionListener for each menu item:

Each menu item (dashboard, parkingSlot, addSlot, deleteSlot, parkedCars, parkCar, deleteParkedCar) has an ActionListener that handles the action event when the respective menu item is clicked. Inside the actionPerformed method of each ActionListener, a specific panel is created and added to the mainPanel. The existing components in the mainPanel are removed, and the mainPanel is refreshed and repainted to display the new panel.

**Dashboard Class**

**GUI Components:**

* JLabels: JLabel is a class of java Swing. JLabel is used to display a short string or an image icon.
  + titleLabel
  + staffLabel
  + visitorLabel
  + totalVisitorParking
  + totalStaffParking
* JTextFields: The object of a JTextField class is a text component that allows the editing of a single line text.
  + totalStaff
  + totalVisitor
* JButton: to create a labeled button.
  + addBtn

**Event Handling Functions:**

actionPerformed(ActionEvent e): This method handles the button click event for the "addBtn" button. It performs various checks and validations on the input data and updates the carPark object accordingly. It also updates the GUI components (totalVisitorParking and totalStaffParking labels) based on the updated values.

**ParkingSlots Class**

**GUI Components:**

* JLabel: Displays the title "Parking Slots" at the top of the panel.
* JPanel:
  + searchPanel: Contains two buttons, "Get Unoccupied Slots" and "Delete Unoccupied Slots," for searching and deleting unoccupied slots.
  + panelTable: Contains a JTable component to display the parking slot information.
* JScrollPane: Wraps the JTable component to provide scrolling functionality.

**Event Handling:**

* ActionListener: The class implements the ActionListener interface to handle button click events.
* actionPerformed method: Overrides the method from the ActionListener interface to perform specific actions based on the button clicked.
  + If the "Get Unoccupied Slots" button is clicked, a DefaultTableModel is created and set as the model for the JTable. Unoccupied slot information is retrieved from the CarPark instance and added to the table.
  + If the "Delete Unoccupied Slots" button is clicked, a confirmation dialog is shown. If confirmed, all unoccupied slots are deleted from the CarPark instance.

Additional Functionality:

* setTb method: Sets the data in the JTable to display all parking slots' information, including slot number, type, occupancy status, car registration number, car owner name, and car parked time. The data is retrieved from the CarPark instance.

**AddSlot Class**

**GUI Components:**

* JLabel: Displays the heading "Add New Slot" at the top of the panel.
* JLabel and JTextField: Allows users to enter the slot identifier.
* JLabel and JComboBox: Allows users to choose the slot type (Staff or Visitor).
* JButton: Triggers the addition of the slot to the CarPark instance.

**Event Handling:**

* ActionListener: The class implements the ActionListener interface to handle button click events.
* actionPerformed method: Overrides the method from the ActionListener interface to perform specific actions based on the button clicked.
  + Checks if the slot identifier and type fields are not empty. If they are, displays an error message.
  + Validates the format of the slot identifier using a regular expression pattern.
  + Checks if the slot identifier already exists in the CarPark instance. If it does, displays an error message.
  + Checks the selected slot type:
    - If "Visitor" is selected, checks if the CarPark instance has reached maximum capacity for visitors. If it hasn't, adds a new ParkingSlot with the provided identifier and "false" (indicating a visitor slot) to the CarPark instance.
    - If "Staff" is selected, checks if the CarPark instance has reached maximum capacity for staff. If it hasn't, adds a new ParkingSlot with the provided identifier and "true" (indicating a staff slot) to the CarPark instance.
  + Displays success messages for adding visitor or staff slots and clears the slot identifier text field.
  + If neither "Visitor" nor "Staff" is selected, displays a message with the combination of the slot identifier and the selected type.
  + Handles any IllegalArgumentException thrown during the process and displays the corresponding error message.

**ParkedCars Class**

**GUI Components:**

* JLabel: Displays the heading "Parked Cars" at the top of the panel.
* JTextField: Allows users to enter a car registration number for searching.
* JButton: Triggers the search for a car registration number.
* JScrollPane: Wraps the JTable component to provide scrollable functionality.
* JTable: Displays the parked cars' information in a tabular format.

**Event Handling:**

* ActionListener: The class implements the ActionListener interface to handle button click events.
* actionPerformed method: Overrides the method from the ActionListener interface to perform specific actions based on the button clicked.
  + Checks if the search field is not empty. If it is, displays an error message.
  + Calls the **findCarByRegistrationNumber** method of the CarPark instance to search for a car with the provided registration number.
  + If the car is not found, displays an error message.
  + If the car is found, updates the table model to display only the information of the searched car, and clears the search field.

**DeleteCar Class**

**GUI Components:**

* JLabel: Displays the heading "Delete Parked Car" at the top of the panel.
* JTextField: Allows users to enter the car registration number for deletion.
* JButton: Triggers the deletion process.
* JOptionPane: Displays confirmation dialogs and error messages.

**Event Handling:**

* ActionListener: The class implements the ActionListener interface to handle button click events.
* actionPerformed method: Overrides the method from the ActionListener interface to perform specific actions based on the button clicked.
  + Checks if the car registration number field is empty. If it is, displays an error message.
  + Shows a confirmation dialog to confirm the deletion of the car.
  + If the user confirms the deletion, calls the **removeCarByRegNumber** method of the CarPark instance to remove the car with the provided registration number.
  + Displays an appropriate success or failure message based on the result of the deletion process.

**ParkCar Class**

**GUI Components:**

* JLabel: Displays the heading "Park New Car" at the top of the panel.
* JTextField: Allows users to enter the car registration number and owner name.
* JComboBox: Allows users to choose the owner type from the options "Staff" or "Visitor".
* JButton: Triggers the process of adding a parked car to the CarPark.
* JOptionPane: Displays error messages and information dialogs.

**Event Handling:**

* ActionListener: The class implements the ActionListener interface to handle button click events.
* actionPerformed method: Overrides the method from the ActionListener interface to perform specific actions based on the button clicked.
  + Validates if all the required fields are filled. If not, displays an error message.
  + Validates the format of the car registration number. If it doesn't match the specified pattern, throws an IllegalArgumentException.
  + Checks if the car with the given registration number is already parked. If it is, displays the parked slot information.
  + If the car is not already parked, creates a new Car instance based on the entered information.
  + Sets the parked time of the car to the current time.
  + Finds the first available parking slot based on the owner type (staff or visitor).
  + If a slot is available, parks the car in the slot and displays a success message.
  + If no slot is available, displays a message indicating the unavailability of slots.
  + Clears the input fields.