**COS70006 Object Oriented Programming**

**Description of GUI component and event handling**

Name: Chaeyeon Im

Student ID: 10532390

1. **Components List**

텍스트, 스크린샷, 소프트웨어, 디자인이(가) 표시된 사진

자동 생성된 설명

Figure 1. Graphic User Interface of parking management system.

The GUI components used in this project are as follows:

• JFrame: Part of the javax.swing package, it is used to create the main window for a application. In Project 2, a window frame with a size of 1280x720 (HD) was created. The layout was then set using GridBagLayout.

• GridBagLayout: This layout was used to precisely adjust the positions of the components.

• GridBagConstraints: Used to control the position, size, and margins of each component. In the code, the properties gridx, gridy, weightx, and weighty are used to impose constraints on the components.

• JButton: A clickable button used for the menu on the left and for creating parking slots.

• JOptionPane: A dialog box used to get input from the user or to display messages to the user.

• ImageIcon: Used to display a car image in the slot when a car is parked.

1. **Event handling list**

• getButtonAddParking(): When the user clicks the "Add a parking spot" button, it prompts the user to enter a slot ID and adds a new parking slot. In the code calls the addSlotPanel(slotID) method to add the new parking slot to the GUI.

• getButtonDeleteParking(): When the user clicks the "Delete a parking spot" button, it prompts the user to enter a slot ID and deletes the specified parking slot. Similar to the previous event handling, it calls the deleteSlotPanel(slotID) method to remove the specified parking slot from the GUI.

• getButtonListParking(): When the user clicks the "List all parking spots" button, it shows the number of occupied and empty parking slots through a JOptionPane component. It also displays the car information in the occupied slots.

• getButtonParkCar(): When the user clicks the "Park a car" button, a dialog box appears via the JOptionPane component where the user can enter the Slot ID, Registration number, Make, Model, and Year of the car they want to park. If entered correctly, the method carParkPanel(slotID, carReg, carMake, carModel, carYear, time) is called to add the car information to the GUI. This method uses the ImageIcon component to add a car image to the specified slot when the car is parked.

• getButtonFindReg(): When the user clicks the "Find car by registration number" button, a dialog box appears via the JOptionPane component. The user can enter the registration number they want to find, and a dialog box with the results appears.

• getButtonRemoveReg(): When the user clicks the "Remove car by registration number" button, a dialog box appears via the JOptionPane component. The user can enter the registration number they want to remove, and a dialog box with the results appears. Then, the method carRemovePanel(carReg) is called to remove the car with the specified registration number from the slot in the GUI. This method also removes the car image from the parking slot using the ImageIcon component.

• getButtonFindMake(): When the user clicks the "Find cars by make" button, a dialog box appears via the JOptionPane component where the user can enter the car make they want to find, and a dialog box with the results appears.

• getButtonReset(): When the user clicks the "Reset car park" button, it calls the carReset() method to remove all parked cars and update the GUI to show all slots as an empty slot.

• getButtonExit(): When the user clicks the "Exit" button, it calls the System.exit() method to terminate the current program.