To analyze and optimize the quarter car model, with a program having a GUI and MVC design pattern, I need to:

1. Create a GUI using QTDesigner and pyuic5,
2. Write a python file containing classes for carModel, carController, and carView
3. Write a car\_App.py file that instantiates a carController object

details for the previously listed steps are in the following paragraphs.

To create the GUI we would first need to put eight QLineEdits with corresponding QLabels inside a QGroupBox with a grid layout that will be nested inside the main windows grid layout. We have the Main Window in a grid layout that way we could use the carView to insert the graph into the grid but not into the QGroupBox. I think we can just use 1 2 spacing for it to be on the right side like shown in the picture provided even though the Main Window would only have 1 sections to start with. We would need to add two QPushButtons for the Calculate and Optimize Suspension functions. We will need to add five QCheckBox with different graphing option shown in the picture provided. The last section inside the QGroupBox is the QLabel where we will display the k min and max along with the SSE. All the parts to this GUI will be in the original QGroupBox besides the Graph which will be added later with the carView.

Next, we will make a python file where we will have all of the behind-the-scenes work done. In order to make the code more legible and easier to handle we will use the Model View Controller Pattern (MVC) which just breaks down the processes into subsections that way it is easier to debug and read. The carModel will be where we store all the information given to the program from the user from the input boxes on the GUI. Then, the carController is where we put all the calculation formulas that will be interacting with the carModel to get the numbers needed to solve the formulas. We will need to find all the formulas required to calculate the X1 and X2 position in the diagram given to use in the word document. The final step is the carView which will be where we have the carController send information we would like to display on the graph and have the carView handle the numbers differently based on the options checked in the QCheckBox contained within the GUI.

Within a file called car\_App.py we will need to have each QLabel, QPushButton, QCheckBox given a name inside the app the controller will be able to call upon and change when needed. This is mostly done by either turning your GUI into a python file or loading the ui with the following code:

form\_class = uic.loadUiType("HW\_8\_GUI.ui")[0]

and then defining the buttons and checkboxes within your Main Window class. Within the QTDesigner we will need to make two new signals and assign them to the two different button presses that way when we press the calculate or Optimize Suspension the carController will know which function inside it that needs to be performed.

With all of this done we will likely have to debug the program to get it to run however it will be much easier as we will be able to narrow it down to one of the sections in the MVC. For example, if the graph is not working, we will know to check the carView first as that is most likely the culprit. There are bound to be some things missed in this document that we will have to find while actually writing the program but this should be a solid basis that will give us or anyone trying to write this program a head start and a place to begin.