```java
package Project4;

import javax.swing.*;
import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.GregorianCalendar;

public class ListEngine extends AbstractListModel {

    private MySingleLinkedList listAutos;
    private MySecondLinkedList undoList;

    public ListEngine() {
        super();
        listAutos = new MySingleLinkedList ();
        undoList = new MySecondLinkedList();
        createList();
    }

    public Auto remove(int i) {
        Auto unit = listAutos.remove(i);
        fireIntervalRemoved(this, 0, listAutos.size());
        return unit;
    }

    public void add (Auto a) {
        listAutos.add(a);
        fireIntervalAdded(this, 0, listAutos.size());
    }

    public Auto get (int i) {
        return listAutos.get(i);
    }

    public Object getElementAt(int arg0) {
        Auto unit = listAutos.get(arg0);
        return unit.toString();
    }

    public int getSize() {
        return listAutos.size();
    }

    public void saveDatabase(String filename) {
        try {
            FileOutputStream fos = new FileOutputStream(filename);
            ObjectOutputStream os = new ObjectOutputStream(fos);
            os.writeObject(listAutos);
            os.close();
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null,"Error in saving db");

        }
    }
```

```
57
58      public void deleteList(){
59          for (int i = 0; i < listAutos.size(); i++)
60              undoList.addU(listAutos.get(i), 2);
61      }
62
63      public void makeUndoList(){
64          for(int i = 0; i < listAutos.size(); i++)
65              undoList.addU(listAutos.get(i), 3);
66      }
67
68      public void loadDatabase(String filename) {
69          deleteList();
70          try {
71              FileInputStream fis = new FileInputStream(filename);
72              ObjectInputStream is = new ObjectInputStream(fis);
73
74              listAutos = (MySingleLinkedList) is.readObject();
75              fireIntervalAdded(this, 0, listAutos.size() - 1);
76              is.close();
77          } catch (Exception ex) {
78              JOptionPane.showMessageDialog(null,"Error in loading db");
79          }
80          makeUndoList();
81          System.out.println("");
82      }
83
84      public void undo(){
85          if (undoList.getTop() == null)
86              return;
87
88          int ROA = undoList.getROA();
89          Auto s = undoList.undo();
90          if (ROA == 1){
91              listAutos.remove(s);
92          }
93          if (ROA == 0){
94              listAutos.add(s);
95          }
96          while (ROA == 3 && listAutos.size() > 0) {
97              listAutos.remove(s);
98              ROA = undoList.getROA();
99              s = undoList.undo();
100         }
101         while (ROA == 2){
102             listAutos.add(s);
103             ROA = undoList.getROA();
104             s = undoList.undo();
105         }
106         fireIntervalAdded(this, 0, listAutos.size());
107         fireIntervalRemoved(this, 0, listAutos.size());
108     }
109
110     public void addTo2ndLinked(Auto s, int ROA){
111         undoList.addU(s, ROA);
112     }
```

```
113
114     public void createList() {
115
116         SimpleDateFormat df = new SimpleDateFormat("MM/dd/yyyy");
117         GregorianCalendar temp1 = new GregorianCalendar();
118         GregorianCalendar temp2 = new GregorianCalendar();
119         GregorianCalendar temp3 = new GregorianCalendar();
120         GregorianCalendar temp4 = new GregorianCalendar();
121         GregorianCalendar temp5 = new GregorianCalendar();
122         GregorianCalendar temp6 = new GregorianCalendar();
123
124         try {
125             Date d1 = df.parse("3/20/2019");
126             temp1.setTime(d1);
127             Date d2 = df.parse("4/20/2019");
128             temp2.setTime(d2);
129             Date d3 = df.parse("12/20/2018");
130             temp3.setTime(d3);
131             Date d4 = df.parse("1/20/2019");
132             temp4.setTime(d4);
133             Date d5 = df.parse("1/20/2010");
134             temp5.setTime(d5);
135             Date d6 = df.parse("1/20/2020");
136             temp6.setTime(d6);
137
138
139             Car Car1 = new Car(temp1, "Outback", 18000,"LX", false);
140             Car Car2 = new Car(temp2, "Chevy", 11000,"EX", false);
141             Car Car3 = new Car(temp3, "Focus", 19000,"EX", true);
142             Truck Truck1 = new Truck(temp4,"F150",12000,"LX",false);
143             Truck Truck2 = new Truck(temp5,"F250",42000,"LX",false);
144             Truck Truck3 = new Truck(temp1,"F350",2000,"EX",true);
145
146             add(Car1);
147             add(Car2);
148             add(Car3);
149             add(Truck1);
150             add(Truck2);
151             add(Truck3);
152
153             addTo2ndLinked(Car1, 1);
154             addTo2ndLinked(Car2, 1);
155             addTo2ndLinked(Car3, 1);
156             addTo2ndLinked(Truck1, 1);
157             addTo2ndLinked(Truck2, 1);
158             addTo2ndLinked(Truck3, 1);
159
160         } catch (ParseException e) {
161             throw new RuntimeException("Error in testing, creation of list");
162         }
163
164     }
165 }
166
```