```java
package inputManagePackage;

import java.io.FileInputStream;

public class InputSystemTxt extends InputSystem {

    protected InputSystemTxt() {
        super();
        minRequiredNumberOfLines = 4;
        generalInfoEndTagLength = 0;
    }

    protected boolean checkEoF(String checkEoF) {
        return false;
    }

    protected void skipLines() {
        // Nothing to skip in the case of Txt
        return;
    }

    /*
     * TODO Consolidate conditional in fact tried simple Consolidate conditional
     */
    protected void inputCvAddSection(int sectionCounter, CVTemplate cvtemplate) throws ParseException {
        String checkNextLine = null;

        if (inputStream.hasNextLine()) {
            checkNextLine = inputStream.nextLine();
        } else {
            checkNextLine = "exit";
        }

        if (sectionRepetitionCheck(checkNextLine, cvtemplate))
            return;

        cvtemplate.addSectionObj(sectionCounter, new Section(checkNextLine));

        if (isSingleParagraphSection(checkNextLine))
            cvtemplate.getSectionObj(sectionCounter).addParagraph(inputStream.nextLine().substring(1));
        else if (hasBulletListItemSection(checkNextLine))
            inputCvAddBulletListItem(sectionCounter, cvtemplate);
        else if (hasBulletListSection(checkNextLine))
            inputCvAddBulletList(sectionCounter, cvtemplate);
        else if (hasNestedBulletListSection(checkNextLine))
            inputCvAddBulletListItemAndItemList(sectionCounter, cvtemplate);
        else if (!checkNextLine.equals("exit"))
            throw new RuntimeException("Invalid Input File " + checkNextLine);

    }

    private boolean hasBulletListSection(String checkNextLine) {
        return checkNextLine.equals("Skills and Experience:");
    }

    private boolean hasNestedBulletListSection(String checkNextLine) {
        return checkNextLine.equals("Professional Experience:");
```

```java
 70     }
 71
 72     private boolean hasBulletListItemSection(String checkNextLine) {
 73         return checkNextLine.equals("Education and Training:") ||
   checkNextLine.equals("Further Courses:")
 74                 || checkNextLine.equals("Career Summary:");
 75     }
 76
 77     private boolean isSingleParagraphSection(String checkNextLine) {
 78         return checkNextLine.equals("Professional Profile:") ||
   checkNextLine.equals("Additional Information:")
 79                 || checkNextLine.equals("Interests:") || checkNextLine.equals("Core
   Strengths:");
 80     }
 81
 82     private void inputCvAddBulletList(int sectionCounter, CVTemplate cvtemplate)
   throws ParseException {
 83         int listCounter = 0;
 84         String checkNextLine = inputStream.nextLine();
 85
 86         while (checkNextLine.substring(0, 2).equals("  ")) {
 87
 88             cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
 89                     new BulletList(checkNextLine.substring(2)));
 90             checkNextLine = inputStream.nextLine();
 91
 92             while (checkNextLine.substring(0, 4).equals("    ")) {
 93                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
 94                         .addBulletListItem(new
   BulletListItem(checkNextLine.substring(4)));
 95                 checkNextLine = inputStream.nextLine();
 96
 97                 if (checkNextLine.isEmpty())
 98                     break;
 99
100             }
101
102             if (checkNextLine.isEmpty())
103                 break;
104             listCounter++;
105
106         }
107         notSkipLine = 1;
108     }
109
110     private void inputCvAddBulletListItem(int sectionCounter, CVTemplate cvtemplate)
   throws ParseException {
111         String checkNextLine = inputStream.nextLine();
112         while (checkNextLine.substring(0, 2).equals("  ")) {
113
114             cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
115                     new BulletListItem(getDate(checkNextLine),
   getWithoutDate(checkNextLine.substring(2))));
116             checkNextLine = inputStream.nextLine();
117
118             if (checkNextLine.isEmpty())
119                 break;
120         }
121         notSkipLine = 1;
```

```java
122     }
123
124     private void inputCvAddBulletListItemAndItemList(int sectionCounter, CVTemplate
    cvtemplate) throws ParseException {
125
126         int listCounter = 0;
127         String checkNextLine = inputStream.nextLine();
128
129         while (checkNextLine.substring(0, 2).equals("  ")) {
130
131             cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
132                     new BulletListItem(getDate(checkNextLine),
    getWithoutDate(checkNextLine.substring(2))));
133             cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter, new
    BulletList(checkNextLine));
134             int listItemCounter = 0;
135             checkNextLine = inputStream.nextLine();
136
137             if (checkNextLine.substring(0, 4).equals("    ")) {
138
139                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
140                         .addBulletListItem(new
    BulletListItem(checkNextLine.substring(4)));
141                 checkNextLine = inputStream.nextLine();
142                 listItemCounter++;
143
144                 if (checkNextLine.isEmpty())
145                     break;
146             } else {
147                 System.out.println("Wrong Input Professional Experience");
148                 break;
149             }
150
151             if (checkNextLine.substring(0, 4).equals("    ")) {
152
153                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
154                         .addBulletListItem(new
    BulletListItem(checkNextLine.substring(4)));
155                 checkNextLine = inputStream.nextLine();
156                 listItemCounter++;
157
158                 if (checkNextLine.isEmpty())
159                     break;
160
161             } else {
162                 System.out.println("Wrong Input Professional Experience");
163                 break;
164             }
165
166             while (checkNextLine.substring(0, 6).equals("      ")) {
167
168                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
169                         .addBulletListItem(new
    BulletListItem(checkNextLine.substring(6)));
170                 checkNextLine = inputStream.nextLine();
171                 listItemCounter++;
172                 if (checkNextLine.isEmpty())
173                     break;
174
```

```java
175              }
176
177          if (checkNextLine.isEmpty())
178              break;
179          listCounter++;
180      }
181      notSkipLine = 1;
182  }
183
184  private boolean sectionRepetitionCheck(String checkNextLine, CVTemplate
    cvtemplate) {
185      for (int i = 0; i < cvtemplate.getNumberOfSectionObj(); i++) {
186          if (checkNextLine.equals(cvtemplate.getSectionObjTitle(i))) {
187              JOptionPane.showMessageDialog(null, "Repeated section of " +
    checkNextLine + ". Skipping.");
188              if (checkNextLine.equals("Professional Profile:") ||
    checkNextLine.equals("Additional Information:")
189                      || checkNextLine.equals("Interests:")) {
190                  inputStream.nextLine();
191
192                  return true;
193              }
194
195          }
196      }
197      return false;
198
199  }
200
201  private Date getDate(String input) throws ParseException {
202      Date date;
203
204      SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
205      date = sdf.parse(input.substring(input.length() - 10));
206
207      return date;
208
209  }
210
211  private String getWithoutDate(String input) {
212
213      return input.substring(0, input.length() - 12);
214  }
215
216  public static void templateTypeIdentifier(CVTemplate cvtemplate) {
217      int checkForFunctional = 0;
218      int checkForChronological = 0;
219      int checkForCombined = 0;
220
221      for (int i = 0; i < cvtemplate.getNumberOfSectionObj(); i++) {
222
223          if (cvtemplate.getSectionObjTitle(i).equals("Skills and Experience:")) {
224              checkForFunctional++;
225              checkForCombined++;
226          } else if (cvtemplate.getSectionObjTitle(i).equals("Career Summary:")) {
227              checkForFunctional++;
228
229          } else if (cvtemplate.getSectionObjTitle(i).equals("Professional
    Experience:")) {
```

```
230                  checkForChronological++;
231                  checkForCombined++;
232              } else if (cvtemplate.getSectionObjTitle(i).equals("Core Strengths:")) {
233                  checkForChronological++;
234
235              }
236
237          }
238
239          if (cvtemplate.getNumberOfSectionObj() < 4)
240              JOptionPane.showMessageDialog(null, "Less than 4 sections.");
241
242          if (checkForChronological != 2 && checkForCombined != 2 && checkForFunctional
    == 2) {
243              JOptionPane.showMessageDialog(null, "Functional CV identified.");
244          } else if (checkForChronological == 2 && checkForCombined != 2 &&
    checkForFunctional != 2) {
245              JOptionPane.showMessageDialog(null, "Chronological CV identified.");
246          } else if (checkForChronological != 2 && checkForCombined == 2 &&
    checkForFunctional != 2) {
247              JOptionPane.showMessageDialog(null, "Combined CV identified.");
248          } else {
249              JOptionPane.showMessageDialog(null, "CV type not identified.");
250          }
251
252      }
253
254      public int checkIfCVFile(String filePath) {
255
256          Scanner fileStream = null;
257          try {
258              fileStream = new Scanner(new FileInputStream(filePath));
259          } catch (FileNotFoundException e2) {
260              // TODO Auto-generated catch block
261              e2.printStackTrace();
262          }
263          int i = 0;
264          while (fileStream.hasNextLine() && fileStream.nextLine().length() > 0) {
265              i++;
266              fileStream.nextLine();
267          }
268
269          if (i < 4) {
270              JOptionPane.showMessageDialog(null, "Invalid input file. Not a CV file.");
271              return 1;
272          }
273          return 0;
274      }
275
276 }
277
```