

InputSystemTxt.java

```

1 package inputManagePackage;
2
3 import java.io.FileInputStream;
16
17 public class InputSystemTxt extends InputSystem {
18
19     protected InputSystemTxt() {
20         super();
21         minRequiredNumberOfLines = 4;
22         generalInfoEndTagLength = 0;
23     }
24
25     protected boolean checkEoF(String checkEoF) {
26         return false;
27     }
28
29     protected void skipLines() {
30         // Nothing to skip in the case of Txt
31         return;
32     }
33
34     /*
35      * TODO Consolidate conditional in fact tried simple Consolidate conditional
36      */
37     protected void inputCvAddSection(int sectionCounter, CVTemplate cvtemplate) throws
    ParseException {
38         String checkNextLine = null;
39
40         if (inputStream.hasNextLine()) {
41             checkNextLine = inputStream.nextLine();
42         } else {
43             checkNextLine = "exit";
44         }
45
46         if (sectionRepetitionCheck(checkNextLine, cvtemplate))
47             return;
48
49         cvtemplate.addSectionObj(sectionCounter, new Section(checkNextLine));
50
51         if (isSingleParagraphSection(checkNextLine))
52             cvtemplate.getSectionObj(sectionCounter).addParagraph(inputStream.nextLine()
    ).substring(1));
53         else if (hasBulletListItemSection(checkNextLine))
54             inputCvAddBulletListItem(sectionCounter, cvtemplate);
55         else if (hasBulletListSection(checkNextLine))
56             inputCvAddBulletList(sectionCounter, cvtemplate);
57         else if (hasNestedBulletListSection(checkNextLine))
58             inputCvAddBulletListItemAndItemList(sectionCounter, cvtemplate);
59         else if (!checkNextLine.equals("exit"))
60             throw new RuntimeException("Invalid Input File " + checkNextLine);
61
62     }
63
64     private boolean hasBulletListSection(String checkNextLine) {
65         return checkNextLine.equals("Skills and Experience:");
66     }
67
68     private boolean hasNestedBulletListSection(String checkNextLine) {
69         return checkNextLine.equals("Professional Experience:");

```

InputSystemTxt.java

```

70     }
71
72     private boolean hasBulletListItemSection(String checkNextLine) {
73         return checkNextLine.equals("Education and Training:") ||
74             checkNextLine.equals("Further Courses:")
75             || checkNextLine.equals("Career Summary:");
76     }
77
78     private boolean isSingleParagraphSection(String checkNextLine) {
79         return checkNextLine.equals("Professional Profile:") ||
80             checkNextLine.equals("Additional Information:")
81             || checkNextLine.equals("Interests:") || checkNextLine.equals("Core
82             Strengths:");
83     }
84
85     private void inputCvAddBulletList(int sectionCounter, CVTemplate cvtemplate)
86     throws ParseException {
87         int listCounter = 0;
88         String checkNextLine = inputStream.nextLine();
89
90         while (checkNextLine.substring(0, 2).equals(" ")) {
91
92             cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
93                 new BulletList(checkNextLine.substring(2)));
94             checkNextLine = inputStream.nextLine();
95
96             while (checkNextLine.substring(0, 4).equals(" ")) {
97                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
98                     .addBulletListItem(new
99                     BulletListItem(checkNextLine.substring(4)));
100                 checkNextLine = inputStream.nextLine();
101
102                 if (checkNextLine.isEmpty())
103                     break;
104             }
105
106             if (checkNextLine.isEmpty())
107                 break;
108             listCounter++;
109         }
110         notSkipLine = 1;
111     }
112
113     private void inputCvAddBulletListItem(int sectionCounter, CVTemplate cvtemplate)
114     throws ParseException {
115         String checkNextLine = inputStream.nextLine();
116         while (checkNextLine.substring(0, 2).equals(" ")) {
117
118             cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
119                 new BulletListItem(getDate(checkNextLine),
120                 getWithoutDate(checkNextLine.substring(2))));
121             checkNextLine = inputStream.nextLine();
122
123             if (checkNextLine.isEmpty())
124                 break;
125         }
126         notSkipLine = 1;

```

```

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

```

InputSystemTxt.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     return false;
197 }
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:")) {

```

InputSystemTxt.java

```

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

```