

InputSystemLatex.java

```

1 package inputManagePackage;
2
3 import java.text.ParseException;
13
14 public class InputSystemLatex extends InputSystem {
15
16     protected InputSystemLatex() {
17         super();
18         minRequiredNumberOfLines = 11;
19         generalInfoEndTagLength = 2;
20     }
21
22     protected boolean checkEoF(String checkEoF) {
23         return checkEoF.equals("\\end{document}");
24     }
25
26     protected void skipLines() {
27         for (int i = 0; i < 5; i++) {
28             inputStream.nextLine();
29         }
30     }
31
32     protected void inputCvAddSection(int sectionCounter, CVTemplate cvtemplate) throws
    ParseException {
33         String checkNextLine = null;
34         String tempStringSection;
35
36         if (inputStream.hasNextLine()) {
37             checkNextLine = inputStream.nextLine();
38         } else {
39             checkNextLine = "exit";
40         }
41
42         if (sectionRepetitionCheck(checkNextLine, cvtemplate))
43             return;
44         checkNextLine = checkNextLine.substring(9, checkNextLine.length() - 1);
45
46         if (checkNextLine.equals("Professional Profile:")) {
47
48             cvtemplate.addSectionObj(sectionCounter, new Section("Professional
    Profile:"));
49             tempStringSection = inputStream.nextLine();
50             cvtemplate.getSectionObj(sectionCounter)
51                 .addParagraph(tempStringSection.substring(0,
    tempStringSection.length() - 2));
52
53             } else if (checkNextLine.equals("Additional Information:")) {
54
55                 cvtemplate.addSectionObj(sectionCounter, new Section("Additional
    Information:"));
56                 tempStringSection = inputStream.nextLine();
57                 cvtemplate.getSectionObj(sectionCounter)
58                     .addParagraph(tempStringSection.substring(0,
    tempStringSection.length() - 2));
59
60             } else if (checkNextLine.equals("Interests:")) {
61
62                 cvtemplate.addSectionObj(sectionCounter, new Section("Interests:"));
63                 tempStringSection = inputStream.nextLine();

```

InputSystemLatex.java

```

64         cvtemplate.getSectionObj(sectionCounter)
65         .addParagraph(tempStringSection.substring(0,
tempStringSection.length() - 2));
66
67     } else if (checkNextLine.equals("Education and Training:")) {
68
69         cvtemplate.addSectionObj(sectionCounter, new Section("Education and
Training:"));
70         inputCvAddBulletListItem(sectionCounter, cvtemplate);
71
72     } else if (checkNextLine.equals("Further Courses:")) {
73
74         cvtemplate.addSectionObj(sectionCounter, new Section("Further Courses:"));
75         inputCvAddBulletListItem(sectionCounter, cvtemplate);
76
77     } else if (checkNextLine.equals("Career Summary:")) {
78
79         cvtemplate.addSectionObj(sectionCounter, new Section("Career Summary:"));
80         inputCvAddBulletListItem(sectionCounter, cvtemplate);
81
82     } else if (checkNextLine.equals("Skills and Experience:")) {
83
84         cvtemplate.addSectionObj(sectionCounter, new Section("Skills and
Experience:"));
85         inputCvAddBulletList(sectionCounter, cvtemplate);
86
87     } else if (checkNextLine.equals("Core Strengths:")) {
88
89         cvtemplate.addSectionObj(sectionCounter, new Section("Core Strengths:"));
90         cvtemplate.getSectionObj(sectionCounter).addParagraph(inputStream.nextLine()
));
91
92     } else if (checkNextLine.equals("Professional Experience:")) {
93
94         cvtemplate.addSectionObj(sectionCounter, new Section("Professional
Experience:"));
95         inputCvAddBulletListItemAndItemList(sectionCounter, cvtemplate);
96
97     } else {
98         if (!checkNextLine.equals("ment"))
99             System.out.println("Invalid Input File " + checkNextLine);
100
101     }
102     sectionCounter++;
103 }
104
105 private void inputCvAddBulletList(int sectionCounter, CVTemplate cvtemplate)
throws ParseException {
106     int listCounter = 0;
107     String checkNextLine = inputStream.nextLine();
108
109     if (checkNextLine.length() < 12 || !checkNextLine.substring(0, 11).equals("\
subsection")) {
110         JOptionPane.showMessageDialog(null, "Invalid input file. Error on
bulletlist.");
111         return;
112     }
113
114     while (checkNextLine.substring(0, 11).equals("\subsection")) {

```

InputSystemLatex.java

```

115
116         cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
117             new BulletList(checkNextLine.substring(12, checkNextLine.length()
118 - 1)));
119         inputStream.nextLine(); // begin itemize
120         checkNextLine = inputStream.nextLine();
121         while (checkNextLine.substring(0, 5).equals("\\item")) {
122             cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
123                 .addBulletListItem(new
124                     BulletListItem(checkNextLine.substring(6)));
125             checkNextLine = inputStream.nextLine();
126         }
127         listCounter++;
128         if (!checkNextLine.equals("\\end{itemize}")) {
129             JOptionPane.showMessageDialog(null, "Invalid input file. Error on
130 bulletlist. No 'enditemize'.");
131             break;
132         }
133         checkNextLine = inputStream.nextLine(); // end itemize
134         if (checkNextLine.length() < 12)
135             break;
136         notSkipLine = 1;
137     }
138 }
139 }
140
141 private void inputCvAddBulletListItem(int sectionCounter, CVTemplate cvtemplate)
142     throws ParseException {
143     String checkNextLine = inputStream.nextLine();
144     if (checkNextLine.equals("\\begin{itemize}"))
145         checkNextLine = inputStream.nextLine();
146     else {
147         JOptionPane.showMessageDialog(null, "Invalid input file. Error on
148 bulletlistitem.");
149         return;
150     }
151     // checkNextLine=inputStream.nextLine();
152     while (checkNextLine.substring(0, 5).equals("\\item")) {
153         cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
154             new BulletListItem(getDate(checkNextLine),
155                 getWithoutDate(checkNextLine.substring(6))));
156         checkNextLine = inputStream.nextLine();
157         if (checkNextLine.length() < 5)
158             break;
159     }
160     if (!checkNextLine.equals("\\end{itemize}")) {
161         JOptionPane.showMessageDialog(null, "Invalid input file. Error on
162 bulletlistitem.");
163         return;
164     }
165 }
166 }

```

```

167
168     private void inputCvAddBulletListItemAndItemList(int sectionCounter, CVTemplate
cvtemplate) throws ParseException {
169
170         int listCounter = 0;
171         String checkNextLine = inputStream.nextLine();
172
173         if (checkNextLine.length() < 12 || !checkNextLine.substring(0, 11).equals("\
\subsection")) {
174             JOptionPane.showMessageDialog(null, "Invalid input file. Error on
bulletlist.");
175             return;
176         }
177
178         while (checkNextLine.substring(0, 11).equals("\subsection")) {
179
180             cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
181                 new BulletListItem(getDate(checkNextLine),
getWithoutDate(checkNextLine.substring(12))));
182             cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
183                 new BulletList(checkNextLine.substring(12, checkNextLine.length()
- 1)));
184             int listItemCounter = 0;
185
186             inputStream.nextLine(); // begin itemize
187             checkNextLine = inputStream.nextLine();
188
189             if (checkNextLine.substring(0, 5).equals("\item")) {
190
191                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
192                     .addBulletListItem(new
BulletListItem(checkNextLine.substring(6)));
193                 checkNextLine = inputStream.nextLine();
194                 listItemCounter++;
195
196                 if (checkNextLine.isEmpty())
197                     break;
198             } else {
199                 System.out.println("Wrong Input Professional Experience");
200                 break;
201             }
202
203             if (checkNextLine.substring(0, 5).equals("\item")) {
204
205                 cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
206                     .addBulletListItem(new
BulletListItem(checkNextLine.substring(6, checkNextLine.length() - 2)));
207                 checkNextLine = inputStream.nextLine();
208                 listItemCounter++;
209
210                 if (checkNextLine.isEmpty())
211                     break;
212             } else {
213                 System.out.println("Wrong Input Professional Experience");
214                 break;
215             }
216
217             while (!checkNextLine.equals("\end{itemize}")) {

```

InputSystemLatex.java

```

219
220         cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
221             .addBulletListItem(new
BulletListItem(checkNextLine.substring(0, checkNextLine.length() - 2)));
222         checkNextLine = inputStream.nextLine();
223         listItemCounter++;
224         if (checkNextLine.isEmpty())
225             break;
226
227     }
228
229     checkNextLine = inputStream.nextLine();
230
231     if (checkNextLine.isEmpty())
232         break;
233     listCounter++;
234
235 }
236 notSkipLine = 1;
237 }
238
239 private boolean sectionRepetitionCheck(String checkNextLine, CVTemplate
cvtemplate) {
240     for (int i = 0; i < cvtemplate.getNumberOfSectionObj(); i++) {
241         if (checkNextLine.equals(cvtemplate.getSectionObjTitle(i))) {
242             JOptionPane.showMessageDialog(null, "Repeated section of " +
checkNextLine + ". Skipping.");
243             if (checkNextLine.equals("Professional Profile:") ||
checkNextLine.equals("Additional Information:")
|| checkNextLine.equals("Interests:")) {
244                 inputStream.nextLine();
245
246                 return true;
247             }
248         }
249     }
250 }
251 }
252 return false;
253
254 }
255
256 private Date getDate(String input) throws ParseException {
257     Date date;
258
259     SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
260     date = sdf.parse(input.substring(input.length() - 10));
261
262     return date;
263 }
264
265
266 private String getWithoutDate(String input) {
267
268     return input.substring(0, input.length() - 12);
269 }
270
271 }
272

```