```java
package inputManagePackage;

import java.io.FileInputStream;

public class InputSystemTxt extends InputSystem {

    protected InputSystemTxt() {
        super();
    }

    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:");
    }

    private boolean hasBulletListItemSection(String checkNextLine) {
        return checkNextLine.equals("Education and Training:") ||
checkNextLine.equals("Further Courses:")
                || checkNextLine.equals("Career Summary:");
    }

    private boolean isSingleParagraphSection(String checkNextLine) {
        return checkNextLine.equals("Professional Profile:") ||
checkNextLine.equals("Additional Information:")
                || checkNextLine.equals("Interests:") || checkNextLine.equals
("Core Strengths:");
    }

    private void inputCvAddBulletList(int sectionCounter, CVTemplate
cvtemplate) throws ParseException {
        int listCounter = 0;
        String checkNextLine = inputStream.nextLine();

        while (checkNextLine.substring(0, 2).equals("  ")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
                    new BulletList(checkNextLine.substring(2)));
            checkNextLine = inputStream.nextLine();

            while (checkNextLine.substring(0, 4).equals("    ")) {
                cvtemplate.getSectionObj(sectionCounter).getBulletList
(listCounter)
                        .addBulletListItem(new BulletListItem
(checkNextLine.substring(4)));
                checkNextLine = inputStream.nextLine();

                if (checkNextLine.isEmpty())
                    break;

            }

            if (checkNextLine.isEmpty())
                break;
            listCounter++;

        }
        notSkipLine = 1;
    }

    private void inputCvAddBulletListItem(int sectionCounter, CVTemplate
cvtemplate) throws ParseException {
        String checkNextLine = inputStream.nextLine();
        while (checkNextLine.substring(0, 2).equals("  ")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
                    new BulletListItem(getDate(checkNextLine), getWithoutDate
(checkNextLine.substring(2))));
```

```
            checkNextLine = inputStream.nextLine();

            if (checkNextLine.isEmpty())
                break;
        }
        notSkipLine = 1;
    }

    private void inputCvAddBulletListItemAndItemList(int sectionCounter,
CVTemplate cvtemplate) throws ParseException {

        int listCounter = 0;
        String checkNextLine = inputStream.nextLine();

        while (checkNextLine.substring(0, 2).equals("  ")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
                    new BulletListItem(getDate(checkNextLine), getWithoutDate
(checkNextLine.substring(2))));
            cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
new BulletList(checkNextLine));
            int listItemCounter = 0;
            checkNextLine = inputStream.nextLine();

            if (checkNextLine.substring(0, 4).equals("    ")) {

                cvtemplate.getSectionObj(sectionCounter).getBulletList
(listCounter)
                        .addBulletListItem(new BulletListItem
(checkNextLine.substring(4)));
                checkNextLine = inputStream.nextLine();
                listItemCounter++;

                if (checkNextLine.isEmpty())
                    break;
            } else {
                System.out.println("Wrong Input Professional Experience");
                break;
            }

            if (checkNextLine.substring(0, 4).equals("    ")) {

                cvtemplate.getSectionObj(sectionCounter).getBulletList
(listCounter)
                        .addBulletListItem(new BulletListItem
(checkNextLine.substring(4)));
                checkNextLine = inputStream.nextLine();
                listItemCounter++;

                if (checkNextLine.isEmpty())
                    break;

            } else {
                System.out.println("Wrong Input Professional Experience");
                break;
            }
```

```
            while (checkNextLine.substring(0, 6).equals("      ")) {

                cvtemplate.getSectionObj(sectionCounter).getBulletList
(listCounter)
                        .addBulletListItem(new BulletListItem
(checkNextLine.substring(6)));
                checkNextLine = inputStream.nextLine();
                listItemCounter++;
                if (checkNextLine.isEmpty())
                    break;

            }

            if (checkNextLine.isEmpty())
                break;
            listCounter++;
        }
        notSkipLine = 1;
    }

    private boolean sectionRepetitionCheck(String checkNextLine, CVTemplate
cvtemplate) {
        for (int i = 0; i < cvtemplate.getNumberOfSectionObj(); i++) {
            if (checkNextLine.equals(cvtemplate.getSectionObjTitle(i))) {
                JOptionPane.showMessageDialog(null, "Repeated section of " +
checkNextLine + ". Skipping.");
                if (checkNextLine.equals("Professional Profile:") ||
checkNextLine.equals("Additional Information:")
                        || checkNextLine.equals("Interests:")) {
                    inputStream.nextLine();

                    return true;
                }

            }
        }
        return false;

    }

    private Date getDate(String input) throws ParseException {
        Date date;

        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        date = sdf.parse(input.substring(input.length() - 10));

        return date;

    }

    private String getWithoutDate(String input) {

        return input.substring(0, input.length() - 12);
    }
```

```
    public static void templateTypeIdentifier(CVTemplate cvtemplate) {
        int checkForFunctional = 0;
        int checkForChronological = 0;
        int checkForCombined = 0;

        for (int i = 0; i < cvtemplate.getNumberOfSectionObj(); i++) {

            if (cvtemplate.getSectionObjTitle(i).equals("Skills and
Experience:")) {
                checkForFunctional++;
                checkForCombined++;
            } else if (cvtemplate.getSectionObjTitle(i).equals("Career
Summary:")) {
                checkForFunctional++;

            } else if (cvtemplate.getSectionObjTitle(i).equals("Professional
Experience:")) {
                checkForChronological++;
                checkForCombined++;
            } else if (cvtemplate.getSectionObjTitle(i).equals("Core
Strengths:")) {
                checkForChronological++;

            }

        }

        if (cvtemplate.getNumberOfSectionObj() < 4)
            JOptionPane.showMessageDialog(null, "Less than 4 sections.");

        if (checkForChronological != 2 && checkForCombined != 2 &&
checkForFunctional == 2) {
            JOptionPane.showMessageDialog(null, "Functional CV identified.");
        } else if (checkForChronological == 2 && checkForCombined != 2 &&
checkForFunctional != 2) {
            JOptionPane.showMessageDialog(null, "Chronological CV
identified.");
        } else if (checkForChronological != 2 && checkForCombined == 2 &&
checkForFunctional != 2) {
            JOptionPane.showMessageDialog(null, "Combined CV identified.");
        } else {
            JOptionPane.showMessageDialog(null, "CV type not identified.");
        }

    }

    public int checkIfCVFile(String filePath) {

        Scanner fileStream = null;
        try {
            fileStream = new Scanner(new FileInputStream(filePath));
        } catch (FileNotFoundException e2) {
            // TODO Auto-generated catch block
            e2.printStackTrace();
        }
        int i = 0;
```

```
        while (fileStream.hasNextLine() && fileStream.nextLine().length() > 0)
{
            i++;
            fileStream.nextLine();
        }

        if (i < 4) {
            JOptionPane.showMessageDialog(null, "Invalid input file. Not a CV
file.");
            return 1;
        }
        return 0;
    }

}
```