```java
package inputManagePackage;

import java.text.ParseException;

public class InputSystemLatex extends InputSystem {

    protected InputSystemLatex() {
        super();
        minRequiredNumberOfLines = 11;
        generalInfoEndTagLength = 2;
    }

    protected boolean checkEoF(String checkEoF) {
        return checkEoF.equals("\\end{document}");
    }

    protected void skipLines() {
        for (int i = 0; i < 5; i++) {
            inputStream.nextLine();
        }
    }

    protected void inputCvAddSection(int sectionCounter, CVTemplate cvtemplate) throws ParseException {
        String checkNextLine = null;
        String tempStringSection;

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

        if (sectionRepetitionCheck(checkNextLine, cvtemplate))
            return;
        checkNextLine = checkNextLine.substring(9, checkNextLine.length() - 1);

        if (checkNextLine.equals("Professional Profile:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Professional Profile:"));
            tempStringSection = inputStream.nextLine();
            cvtemplate.getSectionObj(sectionCounter)
                    .addParagraph(tempStringSection.substring(0, tempStringSection.length() - 2));

        } else if (checkNextLine.equals("Additional Information:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Additional Information:"));
            tempStringSection = inputStream.nextLine();
            cvtemplate.getSectionObj(sectionCounter)
                    .addParagraph(tempStringSection.substring(0, tempStringSection.length() - 2));

        } else if (checkNextLine.equals("Interests:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Interests:"));
            tempStringSection = inputStream.nextLine();
            cvtemplate.getSectionObj(sectionCounter)
                    .addParagraph(tempStringSection.substring(0, tempStringSection.length() - 2));

        } else if (checkNextLine.equals("Education and Training:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Education and Training:"));
            inputCvAddBulletListItem(sectionCounter, cvtemplate);

        } else if (checkNextLine.equals("Further Courses:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Further Courses:"));
            inputCvAddBulletListItem(sectionCounter, cvtemplate);

        } else if (checkNextLine.equals("Career Summary:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Career Summary:"));
            inputCvAddBulletListItem(sectionCounter, cvtemplate);

        } else if (checkNextLine.equals("Skills and Experience:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Skills and Experience:"));
            inputCvAddBulletList(sectionCounter, cvtemplate);

        } else if (checkNextLine.equals("Core Strengths:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Core Strengths:"));
            cvtemplate.getSectionObj(sectionCounter).addParagraph(inputStream.nextLine());

        } else if (checkNextLine.equals("Professional Experience:")) {

            cvtemplate.addSectionObj(sectionCounter, new Section("Professional Experience:"));
            inputCvAddBulletListItemAndItemList(sectionCounter, cvtemplate);

        } else {
            if (!checkNextLine.equals("ment"))
                System.out.println("Invalid Input File " + checkNextLine);

        }
        sectionCounter++;
    }

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

        if (checkNextLine.length() < 12 || !checkNextLine.substring(0, 11).equals("\\subsection")) {
            JOptionPane.showMessageDialog(null, "Invalid input file. Error on bulletlist.");
            return;
        }

        while (checkNextLine.substring(0, 11).equals("\\subsection")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
                    new BulletList(checkNextLine.substring(12, checkNextLine.length() - 1)));
            inputStream.nextLine(); // begin itemize
            checkNextLine = inputStream.nextLine();

            while (checkNextLine.substring(0, 5).equals("\\item")) {
                cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
                        .addBulletListItem(new BulletListItem(checkNextLine.substring(6)));
                checkNextLine = inputStream.nextLine();

            }

            listCounter++;
            if (!checkNextLine.equals("\\end{itemize}")) {
                JOptionPane.showMessageDialog(null, "Invalid input file. Error on bulletlist. No 'enditemize'.");
                break;
            }
            checkNextLine = inputStream.nextLine(); // end itemize
            if (checkNextLine.length() < 12)
                break;
            notSkipLine = 1;
        }

    }

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

        if (checkNextLine.equals("\\begin{itemize}"))
            checkNextLine = inputStream.nextLine();
        else {
            JOptionPane.showMessageDialog(null, "Invalid input file. Error on bulletlistitem.");
            return;
        }
        // checkNextLine=inputStream.nextLine();
        while (checkNextLine.substring(0, 5).equals("\\item")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
                    new BulletListItem(getDate(checkNextLine), getWithoutDate(checkNextLine.substring(6))));
            checkNextLine = inputStream.nextLine();

            if (checkNextLine.length() < 5)
                break;
        }

        if (!checkNextLine.equals("\\end{itemize}")) {
            JOptionPane.showMessageDialog(null, "Invalid input file. Error on bulletlistitem.");
            return;
        }

    }

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

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

        if (checkNextLine.length() < 12 || !checkNextLine.substring(0, 11).equals("\\subsection")) {
            JOptionPane.showMessageDialog(null, "Invalid input file. Error on bulletlist.");
            return;
        }

        while (checkNextLine.substring(0, 11).equals("\\subsection")) {

            cvtemplate.getSectionObj(sectionCounter).addBulletListItem(
                    new BulletListItem(getDate(checkNextLine), getWithoutDate(checkNextLine.substring(12))));
            cvtemplate.getSectionObj(sectionCounter).addBulletList(listCounter,
                    new BulletList(checkNextLine.substring(12, checkNextLine.length() - 1)));
            int listItemCounter = 0;

            inputStream.nextLine(); // begin itemize
            checkNextLine = inputStream.nextLine();

            if (checkNextLine.substring(0, 5).equals("\\item")) {

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

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

            if (checkNextLine.substring(0, 5).equals("\\item")) {

                cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
                        .addBulletListItem(new BulletListItem(checkNextLine.substring(6, checkNextLine.length() - 2)));
                checkNextLine = inputStream.nextLine();
                listItemCounter++;

                if (checkNextLine.isEmpty())
                    break;

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

            while (!checkNextLine.equals("\\end{itemize}")) {

                cvtemplate.getSectionObj(sectionCounter).getBulletList(listCounter)
                        .addBulletListItem(new BulletListItem(checkNextLine.substring(0, checkNextLine.length() - 2)));
                checkNextLine = inputStream.nextLine();
                listItemCounter++;
                if (checkNextLine.isEmpty())
                    break;

            }

            checkNextLine = inputStream.nextLine();

            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);
    }

}

```