**Design Goals and Objectives**

The primary objective of the assembler is to translate instructions into octal base representations. The assembler takes an input file and generates two output files: a listing file and a text file. The listing file comprises memory locations in the first column and the octal base representation of the instructions in the second column. Additionally, the listing file includes the assembly code of the instructions in the third column.

**Architecture Overview**

**A screenshot of a computer

Description automatically generated**

**Figure 1:** UML Class Diagram Of the Project

As can be seen above in Figure 1, we created a separate File Manager for handling file-related tasks such as reading from a file or writing to a file. We use an assembler to handle all the logic about converting instructions to octal base representations. The assembler class has a locToInstructionMap, in which we keep memory location information as the key and the instruction information as the value. This way, we mapped each location to the corresponding instruction. FileManager reads every line in the input file and separates each line based on space character. Additionally, we had to iterate the entire file in a separate method to get the final location information before calling the assembler. Using the opCode, remainingFields, and finalLocation for LOC instruction as parameters, later we call our assembler and the assembler handles the remaining logic. At the end of the operation, we call the create load file and create listing file methods from the FileManager, to create two output files.

**Testing and Validation**

For testing and validating our results, we used three input.txt files, which were converted to related load.txt, and listing.txt files. We manually converted each instruction to its corresponding octal-based representations and checked if our results and program outputs were matching. While testing the input file that is provided in the project description, we realized that there was an error in the octal representation of JZ 1,0, which was provided as 013100. Based on our results, the correct octal representation for this instruction should be 014100.