

Assignment9 Design Report

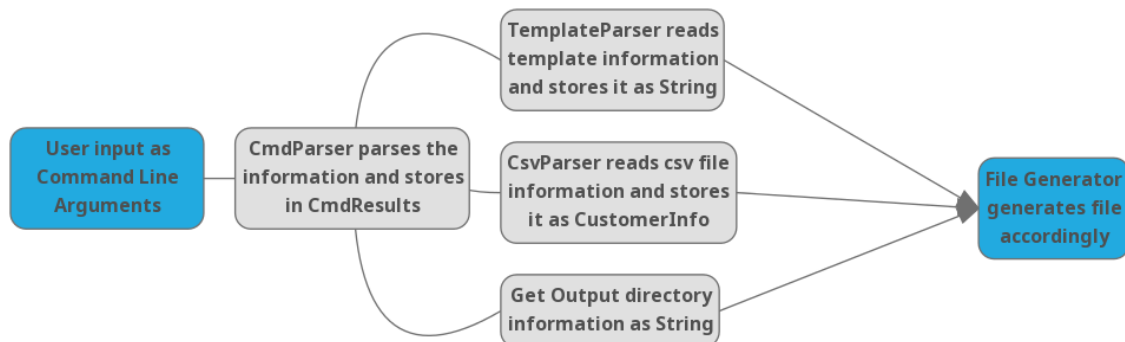
1. General Introduction

Our design mainly includes the following functional classes:

class	Description
CmdParser	parse the command line arguments the user inputs
CsvParser	parse the csv file based on the user input
TemplateParser	parse either the email or the letter template based on the user input
FileGenerator	based on the template information and the command line argument information, generate the email or letter in the output directory
AutoSystem	integrate these functions together; this is the entry point of the java program and enables users to type in command line arguments

In addition, we have several classes designed for storing the intermediate information produced by some classes, including CmdResult, CustomerInfo, Option and Options.

2. Work Flow:



3. Handling special requirements:

- Our program can handle cases when there's comma enclosed in double quotes in the csv files
- After we read the csv file, we return two pieces of information: a HashMap which maps the header with the corresponding column number (for example: "first_name":0), and a list of String arrays which stores the customer information line by line. This design enables our program to work with any csv files and templates.
- We only read the template file once with a StringBuilder and store that as a string to improve the efficiency.
- We create an Option class to represent the flag, in which client can define the attribute of this option such as is it required or optional and how many arguments does it need. Then we put all the option into bundle and sent it to the CmdParser.
- The CmdParser return a result class which contains all the options and their corresponding arguments. It is also responsible for checking the validity of the parsed options.