**Assignment9 Design Report**

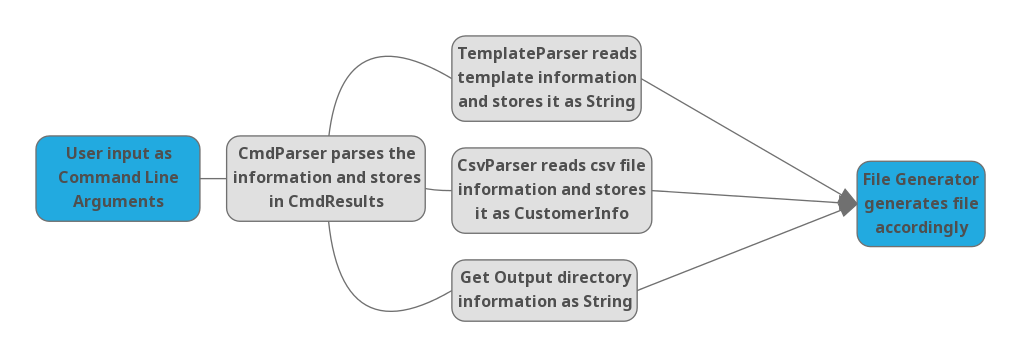
1. General Introduction

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 returns 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.
* When parsing the command line arguments, we use the Option class to record if a flag(for example “--email”) is given in the command line argument.